Data Cleaning Example

jb

5/20/2020

Intro

Lets start this exercise with loading some student admissions data. This is a simple example where we will explore our data – no other real goal.

```
library(ggplot2)
##
## DATA SET
##
##
##
Myfile="SummerStudentAdmissions3_.csv"
## USE YOUR OWN PATH AS NEEDED
MyData <- read.csv(Myfile)
```

Data Acquisition and Data Cleaning

After loading the data, its a good idea to view it to confirm that the data loaded correctly. Try using commands "View", "str" or "head".

```
############################################
## Part 1: Cleaning the data
##         using data vis - ggplot
##
##         EDA is Exploratory Data ANalysis
##             Clean and explore...
################################################

## LOOK AT Each Variable.
str(MyData)
```

```
## 'data.frame':    88 obs. of  9 variables:
##  $ Decision      : Factor w/ 5 levels "","Admit","Banana",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Gender        : Factor w/ 3 levels "","Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ DateSub       : Factor w/ 73 levels "1/10/2020","1/11/2020",..: 2 2 3 40 32 37 41 23 15 5 ...
##  $ State         : Factor w/ 12 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
##  $ GPA           : num  3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
##  $ WorkExp       : num  0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
##  $ TestScore     : int  965 962 969 969 967 956 969 799 969 969 ...
##  $ WritingScore  : int  11 97 93 97 94 89 94 97 93 99 ...
##  $ VolunteerLevel: int  1 0 0 2 2 1 2 5 0 4 ...
```

```
## Notice that there are 9 variables

## Variable (also called features, attributes, columns) Name
(MyVarNames<-names(MyData))
```

```
## [1] "Decision"       "Gender"         "DateSub"        "State"
## [5] "GPA"            "WorkExp"        "TestScore"      "WritingScore"
## [9] "VolunteerLevel"
```

```
MyVarNames[1]
```

```
## [1] "Decision"
```

```
MyData[MyVarNames[1]]
```

```
##    Decision
## 1     Admit
## 2     Admit
## 3     Admit
## 4     Admit
## 5     Admit
## 6     Admit
## 7     Admit
## 8     Admit
## 9     Admit
## 10    Admit
## 11    Admit
## 12    Admit
## 13    Admit
## 14    Admit
## 15    Admit
## 16    Admit
## 17    Admit
## 18    Admit
## 19   Banana
## 20  Decline
## 21  Decline
## 22  Decline
## 23  Decline
## 24  Decline
## 25  Decline
## 26  Decline
## 27  Decline
## 28  Decline
## 29  Decline
## 30  Decline
## 31  Decline
## 32  Decline
## 33  Decline
## 34  Decline
## 35  Decline
## 36  Decline
## 37 Waitlist
## 38 Waitlist
## 39 Waitlist
## 40 Waitlist
## 41 Waitlist
## 42 Waitlist
## 43 Waitlist
## 44 Waitlist
## 45 Waitlist
## 46 Waitlist
## 47 Waitlist
## 48 Waitlist
## 49 Waitlist
## 50 Waitlist
## 51 Waitlist
## 52 Waitlist
## 53 Waitlist
## 54 Waitlist
## 55 Waitlist
## 56 Waitlist
## 57
## 58    Admit
## 59    Admit
## 60    Admit
## 61    Admit
## 62    Admit
## 63    Admit
## 64    Admit
## 65    Admit
## 66    Admit
## 67    Admit
## 68    Admit
## 69    Admit
## 70    Admit
## 71    Admit
## 72    Admit
## 73  Decline
## 74  Decline
## 75  Decline
## 76  Decline
## 77  Decline
## 78  Decline
## 79  Decline
## 80  Decline
## 81  Decline
## 82  Decline
## 83 Waitlist
## 84 Waitlist
## 85 Waitlist
## 86 Waitlist
## 87 Waitlist
## 88  Decline
```

```
(NumColumns <-ncol(MyData))
```

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

```
View(MyData)
```

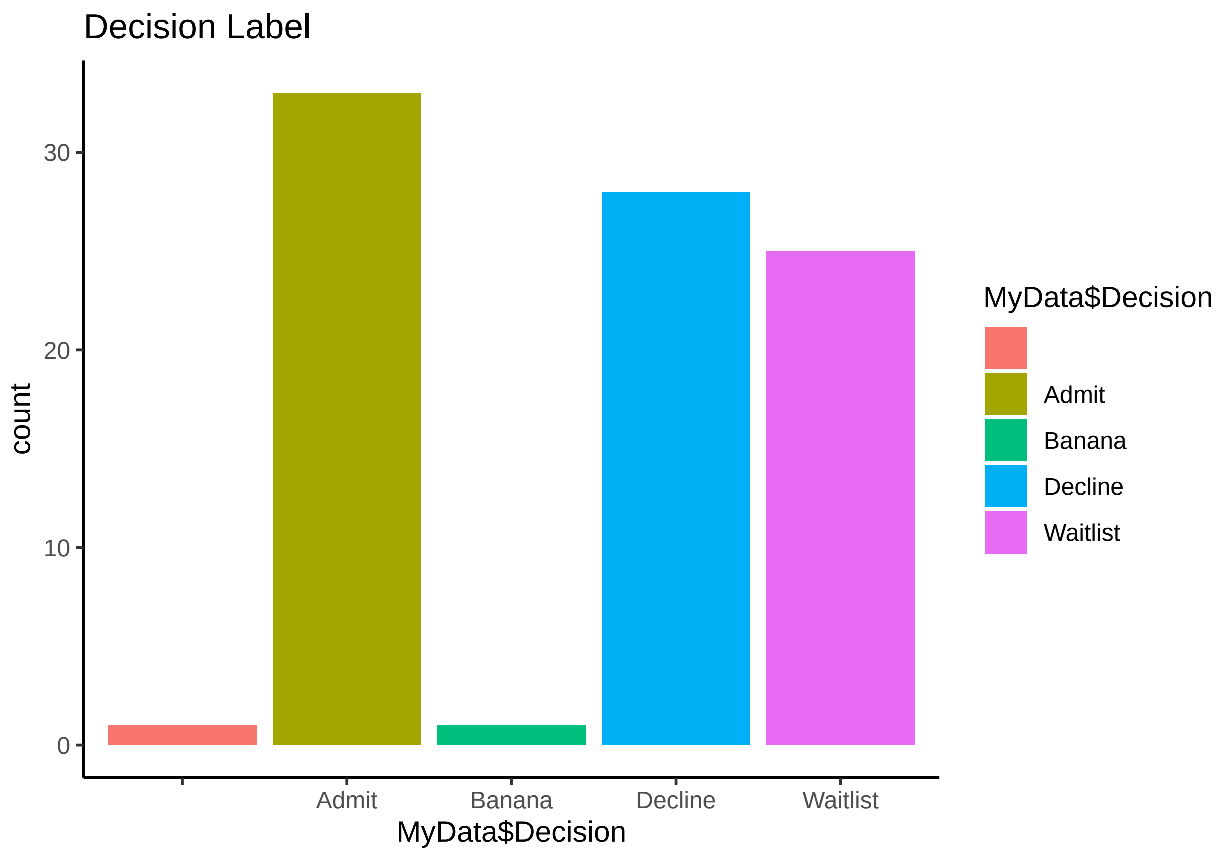
Note that the "label" is the first column in the data frame. This is standard in R. The label is the class or classification of the data (often the dependent variable). Thus not considered part of the data, but rather the label. This variable should be of type factor, so lets confirm.

```
##############################
## Column 1: Decision
###################################

## THis is NOT part of the data!
## It is the LABEL of the data.

## Dataset labels should be of type factor
str(MyData$Decision)
```

```
##  Factor w/ 5 levels "","Admit","Banana",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
## VISUALIZE to SEE what/where the errors are
theme_set(theme_classic())
MyBasePlot1 <- ggplot(MyData)
(MyBasePlot1<-MyBasePlot1 +
    geom_bar(aes(MyData$Decision, fill = MyData$Decision)) +
    ggtitle("Decision Label"))
```

```
## Warning: Use of `MyData$Decision` is discouraged. Use `Decision` instead.

## Warning: Use of `MyData$Decision` is discouraged. Use `Decision` instead.
```



Uncovering Issues

OK - We have problems. Upon inspection of this one column … - 1) We have a blank level - likely from a missing value. - 2) We have a label called banana - whichis wrong.??!?

Fixing Issues

Let's fix these. To fix factor data, first convert it to char. Lets remove "invalid rows", and confirm via inspection.

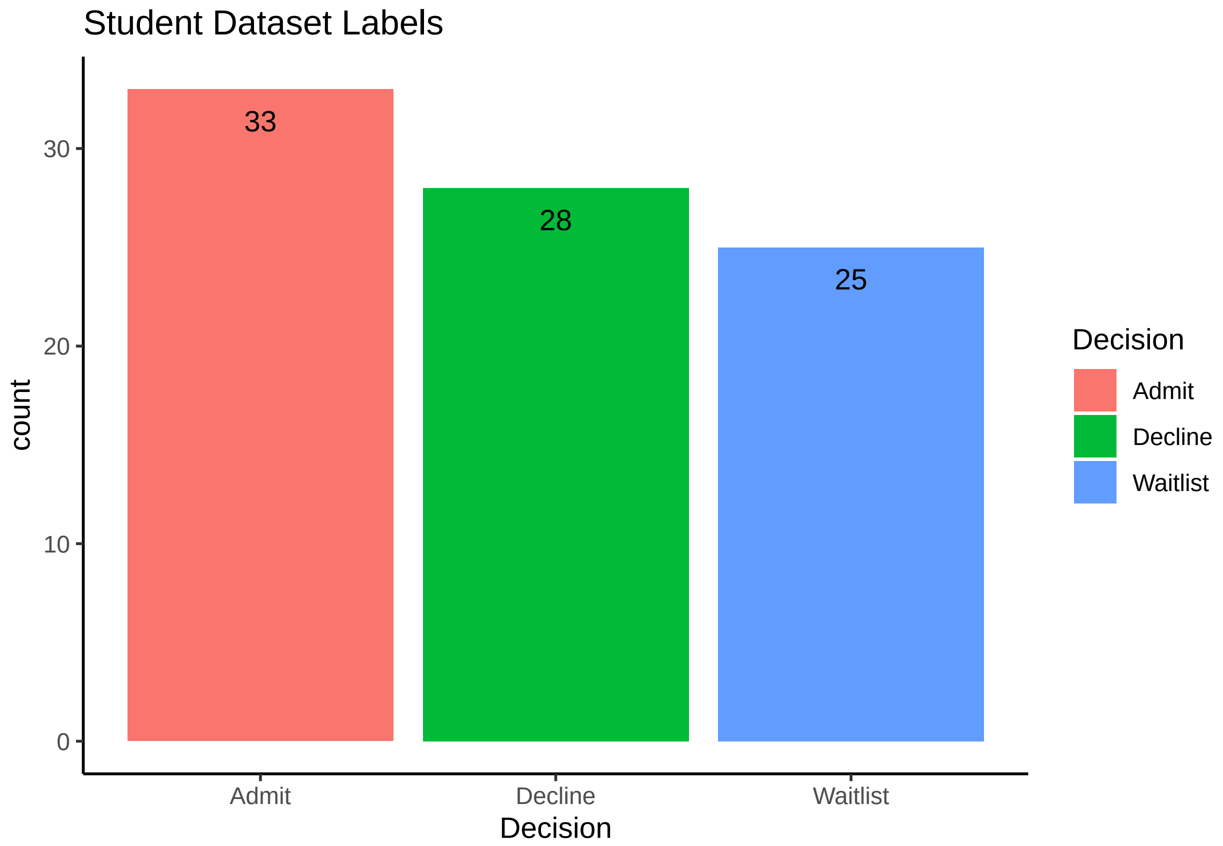
```
nrow(MyData)
```

```
## [1] 88
```

```
MyData$Decision <- as.character(MyData$Decision)

## Keep only rows that are  "Admit", "Decline", or "Waitlist"

MyData <- MyData[(MyData$Decision == "Admit" |
                  MyData$Decision == "Decline" |
                  MyData$Decision == "Waitlist"),]

nrow(MyData)
```

```
## [1] 86
```

```
## Check it again

(MyPlot1<-ggplot(MyData, aes(x=Decision, fill=Decision)) +
    geom_bar()+
    geom_text(stat='count',aes(label=..count..),vjust=2)+
    ggtitle("Student Dataset Labels"))
```



More Cleaning . . . .

Success! Now we can see (and show others) that theLabel in the dataset it clean and balanced. NOTE that we have color, a title, an x-axis label and labeled bars. We also have a legend.

We are not done!! We need to change Decision back to a factor and inspect the other variables.

```
(str(MyData$Decision))
```

```
##  chr [1:86] "Admit" "Admit" "Admit" "Admit" "Admit" "Admit" "Admit" "Admit" ...
```

```
## NULL
## This needs to be changed to type: factor
MyData$Decision<-as.factor(MyData$Decision)
## Check it
table(MyData$Decision)
```

```
##
##    Admit  Decline Waitlist
##       33       28       25
```

```
str(MyData$Decision)
```

```
##  Factor w/ 3 levels "Admit","Decline",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
## Good! We now have factor data with 3 levels.
```

Lets look at Gender next! This is a qualitative variable, lets visualize using a pie chart.

```
#################################################
## THe  next variable to look at is Gender
## Like Decision, Gender is also qualitative.
## Let's use a pie to look at it...
#################################################

str(MyData$Gender)
```

```
##  Factor w/ 3 levels "","Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
```

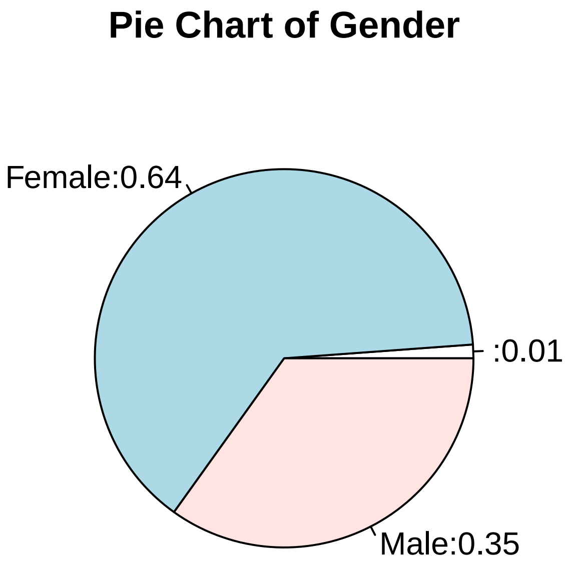
```
NumRows=nrow(MyData)
(TempTable <- table(MyData$Gender))
```

```
##
##        Female   Male
##      1     55     30
```

```
(MyLabels <- paste(names(TempTable), ":",
                   round(TempTable/NumRows,2)  ,sep=""))
```

```
## [1] ":0.01"       "Female:0.64" "Male:0.35"
```

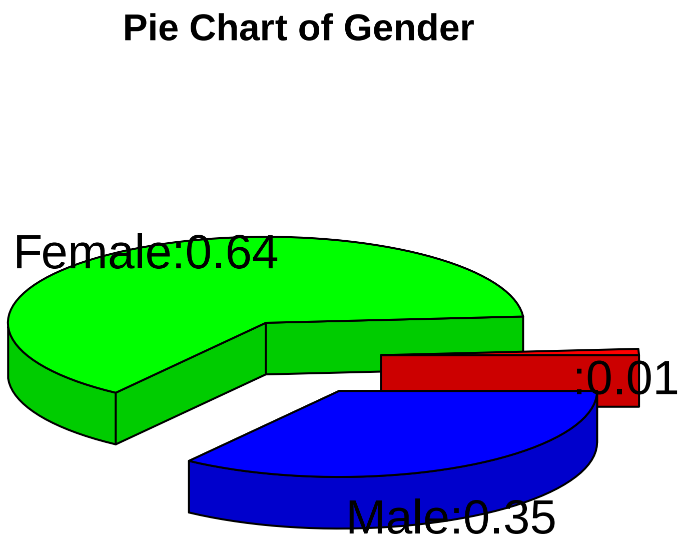
```
pie(TempTable, labels = MyLabels,
    main="Pie Chart of Gender")
```



```
#install.packages("plotrix")
library(plotrix) # Cool 3-d plot here!!
```

```
## Warning: package 'plotrix' was built under R version 3.5.3
```

```
pie3D(TempTable,labels=MyLabels,explode=0.3,
      main="Pie Chart of Gender ")
```



```
table(MyData$Gender)
```

```
##
##         Female   Male
##       1     55     30
```

Houston … We have one problem! We have a blank or NA in the data … but how to fix this? Lets use "is.na"

```
(sum(is.na(MyData$Gender)))  ## This confirms that it is not NA
```

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

Interesting … our mystery value is not an "NA" … what is it??

```
## Let's look at str
str(MyData$Gender)
```

```
##  Factor w/ 3 levels "","Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
```

```
## This shows that we have blank and not NA....
## FIX - change to char, correct, change back to factor
## Keep track of what you are removing from the dataset
```

Its a “blank”. Lets get rid of this row.

```
nrow(MyData)
```

```
## [1] 86
```

```
MyData$Gender <- as.character(MyData$Gender)
## Keep only rows that are Male or Female

MyData <- MyData[(MyData$Gender == "Male" |
                    MyData$Gender == "Female") ,]
nrow(MyData)
```

```
## [1] 85
```

```
## Turn back to factor
MyData$Gender<- as.factor(MyData$Gender)
str(MyData$Gender)
```

```
##  Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
```

```
table(MyData$Gender)
```

```
##
## Female   Male
##     55     30
```

Lets recreate our Data Viz to confirm!

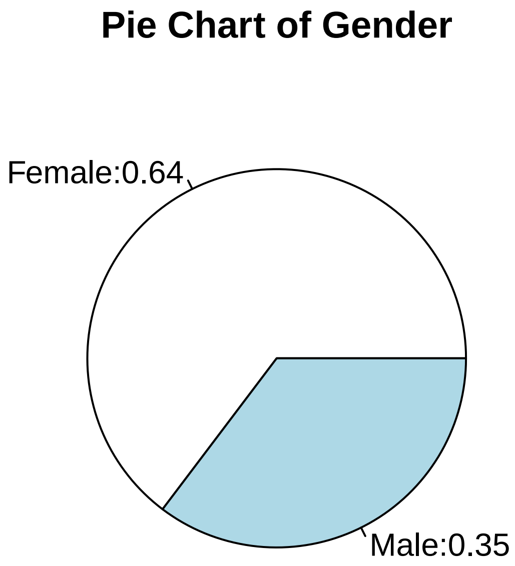
```
(TempTable <- table(MyData$Gender))
```

```
##
## Female   Male
##     55     30
```

```
(MyLabels <- paste(names(TempTable), ":",
                   round(TempTable/NumRows,2) ,sep=""))
```

```
## [1] "Female:0.64" "Male:0.35"
```

```
pie(TempTable, labels = MyLabels,
    main="Pie Chart of Gender")
```



Lets inspect and clean the remaining variables.

```
############################################
## Next variable is: DateSub
#############################################
#names(MyData)
## Check format
str(MyData$DateSub)  ## It is incorrect.
```

```
##  Factor w/ 73 levels "1/10/2020","1/11/2020",..: 2 2 3 40 32 37 41 23 15 5 ...
```

```
## Check for NAs
(sum(is.na(MyData$DateSub)))
```

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

```
## Check the table
table(MyData$DateSub)
```

```
##
##  1/10/2020  1/11/2020  1/12/2020  1/14/2020  1/15/2020  1/17/2020  1/22/2020
##          2          2          3          1          2          1          1
##  1/23/2020  1/25/2020  1/28/2020  1/29/2020  1/30/2020  1/31/2020   1/5/2020
##          1          2          1          1          1          1          2
## 10/10/2019 10/14/2019 10/19/2019 10/25/2019  10/3/2019 10/30/2019 10/31/2019
##          1          1          1          1          1          1          1
##  10/4/2019  10/7/2019  11/1/2019 11/10/2019 11/15/2019 11/16/2019 11/17/2019
##          0          1          1          1          1          1          1
## 11/18/2019 11/19/2019  11/2/2019 11/21/2019 11/25/2019 11/26/2019 11/27/2019
##          0          1          1          1          1          1          1
## 11/28/2019  11/3/2019 11/30/2019  11/4/2019  11/7/2019  11/8/2019  11/9/2019
##          1          1          1          2          2          1          1
##  12/1/2019 12/10/2019 12/11/2019 12/20/2019 12/21/2019 12/23/2019 12/24/2019
##          1          1          1          2          1          1          1
## 12/25/2019 12/27/2019 12/28/2019 12/29/2019  12/3/2019 12/30/2019 12/31/2019
##          1          1          1          1          2          2          1
##  12/4/2019  12/6/2019  12/8/2019  12/9/2019   2/1/2020  2/10/2020  2/11/2020
##          2          1          1          1          2          1          1
##  2/15/2020  2/16/2020  2/19/2020   2/2/2020  2/27/2020   2/4/2020   2/7/2020
##          2          1          1          0          1          1          1
##  3/20/2020   3/6/2020  9/13/2019
##          1          1          1
```

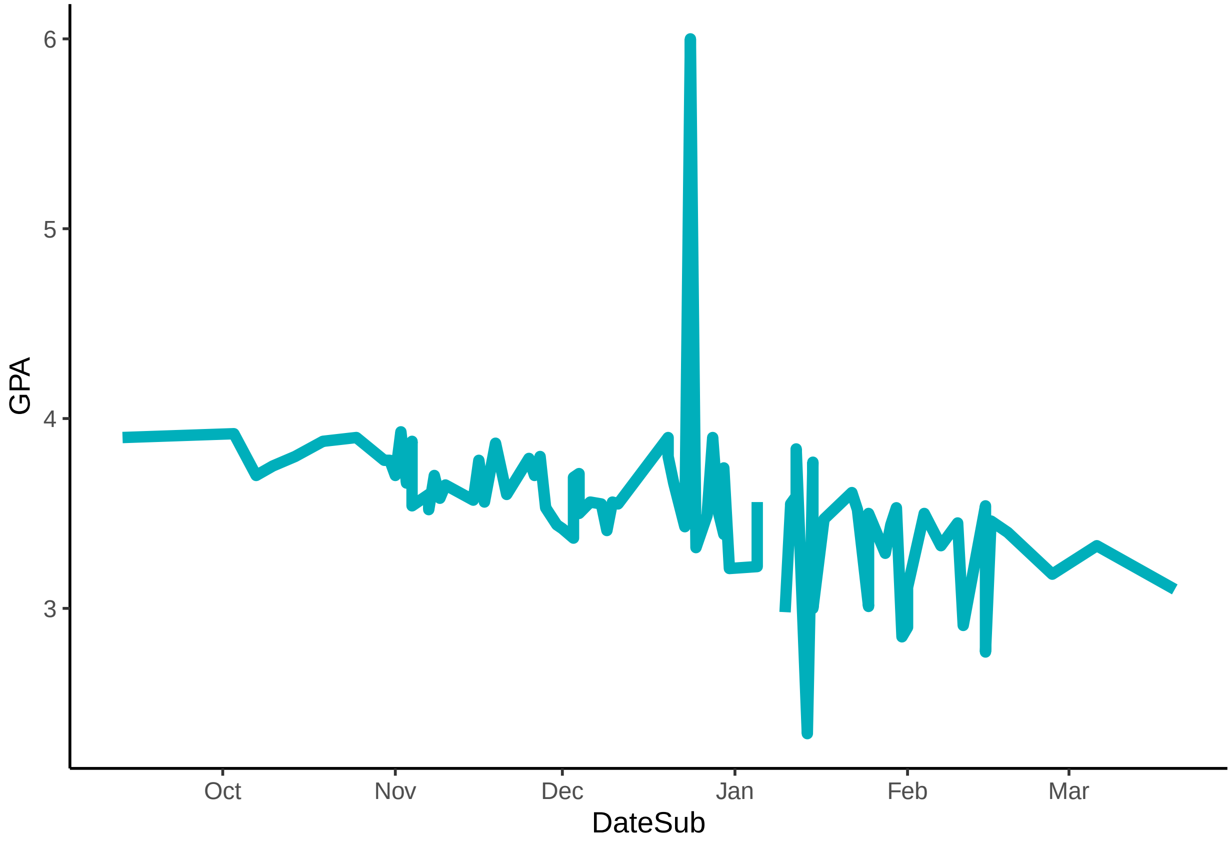
```
## The dates look ok - but the format is wrong and
## needs to be DATE
(MyData$DateSub <- as.Date(MyData$DateSub, "%m/%d/%Y") )
```

```
##  [1] "2020-01-11" "2020-01-11" "2020-01-12" "2019-11-07" "2019-11-21"
##  [6] "2019-11-03" "2019-11-08" "2019-10-07" "2019-10-10" "2020-01-15"
## [11] "2019-10-31" "2019-10-30" "2019-10-14" "2019-11-04" "2019-12-20"
## [16] "2019-10-25" "2019-12-28" "2020-01-10" "2020-01-14" "2020-01-31"
## [21] "2020-01-10" "2020-01-25" "2020-02-27" "2019-12-31" "2020-03-06"
## [26] "2020-02-07" "2019-12-03" "2019-11-30" "2020-01-12" "2020-02-15"
## [31] "2019-12-10" "2020-01-22" "2019-12-04" "2019-11-25" "2020-01-12"
## [36] "2019-12-30" "2020-02-19" "2019-12-09" "2019-12-23" "2020-01-29"
## [41] "2020-02-16" "2020-01-17" "2019-12-27" "2020-02-04" "2019-12-04"
## [46] "2020-01-23" "2020-01-30" "2019-11-28" "2019-11-04" "2019-12-08"
## [51] "2019-12-11" "2019-12-06" "2019-11-17" "2019-11-15" "2019-11-09"
## [56] "2020-01-25" "2019-11-10" "2019-12-21" "2019-12-03" "2019-11-26"
## [61] "2019-11-01" "2019-11-16" "2019-12-20" "2019-11-27" "2019-11-19"
## [66] "2019-10-19" "2019-09-13" "2019-10-03" "2019-11-02" "2019-12-24"
## [71] "2020-02-15" "2020-02-01" "2020-02-11" "2020-01-15" "2020-03-20"
## [76] "2020-02-01" "2020-01-05" "2019-12-25" "2020-01-05" "2019-12-30"
## [81] "2020-01-28" "2019-12-01" "2020-02-10" "2019-12-29" "2019-11-07"
```

```
str(MyData$DateSub)
```

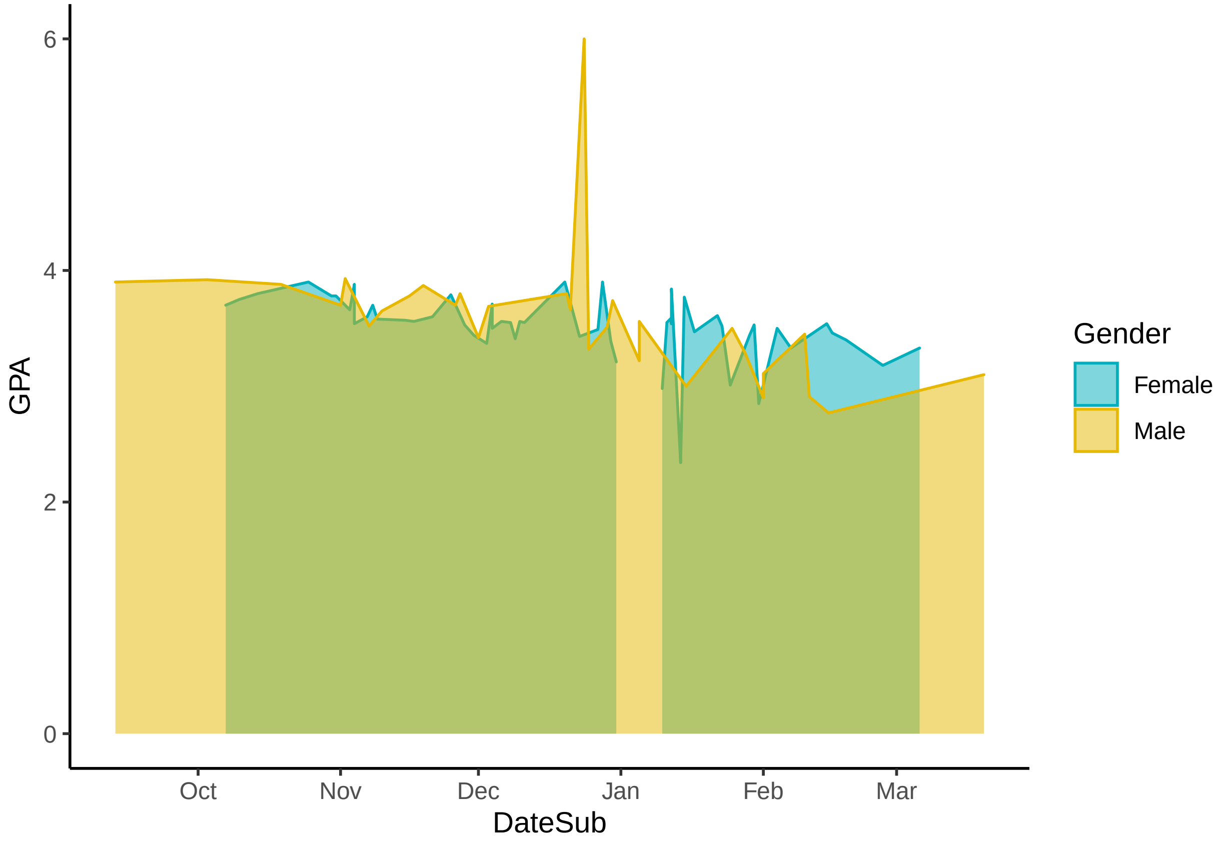
```
##  Date[1:85], format: "2020-01-11" "2020-01-11" "2020-01-12" "2019-11-07" "2019-11-21" ...
```

```
## NOw that we have dates, can visualize them with
## a time series vis option.

ggplot(data = MyData, aes(x = DateSub, y = GPA))+
  geom_line(color = "#00AFBB", size = 2)
```



GPA … above 4.0 …. ?

```
## We have a problem!
## The GPA should never be above 4.0.

ggplot(MyData, aes(x = DateSub, y = GPA)) +
  geom_area(aes(color = Gender, fill = Gender),
            alpha = 0.5, position = position_dodge(0.8)) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  scale_fill_manual(values = c("#00AFBB", "#E7B800"))
```



```
## We can already SEE many things.
## We can see that Males applied a bit early and a bit later.
## We can see that we have an error in at least one GPA
## value that we will need to fix.
## We can see that Female and Male application times and GPAs
## do not appear sig diff - but we can investigate this further.
```

Let's look at GPA and then dates with it

```
str(MyData$GPA)
```

```
##  num [1:85] 3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
```

```
MyData$GPA<-as.numeric(MyData$GPA)
table(MyData$GPA)
```

```
## 
## 2.34 2.77 2.85  2.9 2.91 2.98    3 3.01  3.1 3.11 3.18 3.21 3.22 3.29 3.32 3.33 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2 
## 3.37 3.39  3.4 3.41 3.42 3.43 3.44 3.45 3.46 3.47 3.49  3.5 3.51 3.52 3.53 3.54
##    1    1    1    1    1    1    2    1    1    1    1    3    1    2    2    4
## 3.55 3.56 3.57 3.58 3.59  3.6 3.61 3.65 3.66 3.69  3.7 3.71 3.74 3.75 3.77 3.78
##    3    4    1    1    1    2    1    1    2    1    4    1    1    1    1    3
## 3.79  3.8 3.84 3.87 3.88  3.9 3.92 3.93    6
##    1    3    1    1    2    4    1    1    1
```

```
## Are there NAs?
(sum(is.na(MyData$GPA)))
```

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

```
## Fix the missing GPA first
## Find it
(MissingGPA <- MyData[is.na(MyData$GPA),])
```

```
##    Decision Gender    DateSub      State GPA WorkExp TestScore WritingScore
## 18    Admit Female 2020-01-10 California  NA     2.8       967           95
##    VolunteerLevel
## 18              3
```

```
## OK - its a Female/Admit. We can replace the missing GPA
## with the median of all Female Admits.
(Temp<-MyData[MyData$Decision=="Admit" & MyData$Gender=="Female",])
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1     Admit Female 2020-01-11    Florida 3.54     0.7       965           11
## 2     Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3     Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4     Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5     Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6     Admit Female 2019-11-03 California 3.66     0.9       956           89
## 7     Admit Female 2019-11-08 California 3.70     1.2       969           94
## 8     Admit Female 2019-10-07 California 3.70     2.7       799           97
## 9     Admit Female 2019-10-10   Colorado 3.75     1.1       969           93
## 10    Admit Female 2020-01-15    Florida 3.77     1.4       969           99
## 11    Admit Female 2019-10-31 California 3.78     8.7       966           91
## 12    Admit Female 2019-10-30       Utah 3.78     1.2       968           87
## 13    Admit Female 2019-10-14    Florida 3.80     1.9       965           94
## 14    Admit Female 2019-11-04   Colorado 3.88     1.0       969           93
## 15    Admit Female 2019-12-20    Florida 3.90     4.7       961           93
## 16    Admit Female 2019-10-25   Colorado 3.90     3.8       967           98
## 17    Admit Female 2019-12-28    Florida 3.90     0.0       967           88
## 18    Admit Female 2020-01-10 California   NA     2.8       967           95
##    VolunteerLevel
## 1               1
## 2               0
## 3               0
## 4               2
## 5               2
## 6               1
## 7               2
## 8               5
## 9                0
## 10               4
## 11               2
## 12               2
## 13               5
## 14               4
## 15               1
## 16               3
## 17               0
## 18               3
```

```
## The median for Female Admits is:
(MyMed<-median(Temp$GPA, na.rm=TRUE))
```

```
## [1] 3.75
```

```
## NOW - replace the missing GPA with this Median
MyData$GPA[is.na(MyData$GPA)] <- MyMed
## Check to assure the missing value  was updated...
(sum(is.na(MyData$GPA)))
```

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

```
table(MyData$GPA)
```

```
## 
## 2.34 2.77 2.85  2.9 2.91 2.98    3 3.01  3.1 3.11 3.18 3.21 3.22 3.29 3.32 3.33 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2 
## 3.37 3.39  3.4 3.41 3.42 3.43 3.44 3.45 3.46 3.47 3.49  3.5 3.51 3.52 3.53 3.54 
##    1    1    1    1    1    1    2    1    1    1    1    3    1    2    2    4 
## 3.55 3.56 3.57 3.58 3.59  3.6 3.61 3.65 3.66 3.69  3.7 3.71 3.74 3.75 3.77 3.78 
##    3    4    1    1    1    2    1    1    2    1    4    1    1    2    1    3 
## 3.79  3.8 3.84 3.87 3.88  3.9 3.92 3.93    6 
##    1    3    1    1    2    4    1    1    1
```

Well – the dilema faced by data scientists everywhere … what to do with missing data?!? Its common to either remove the row (as we have done previously); or we can try to replace the value with an estimate – like the mean or median estimate.

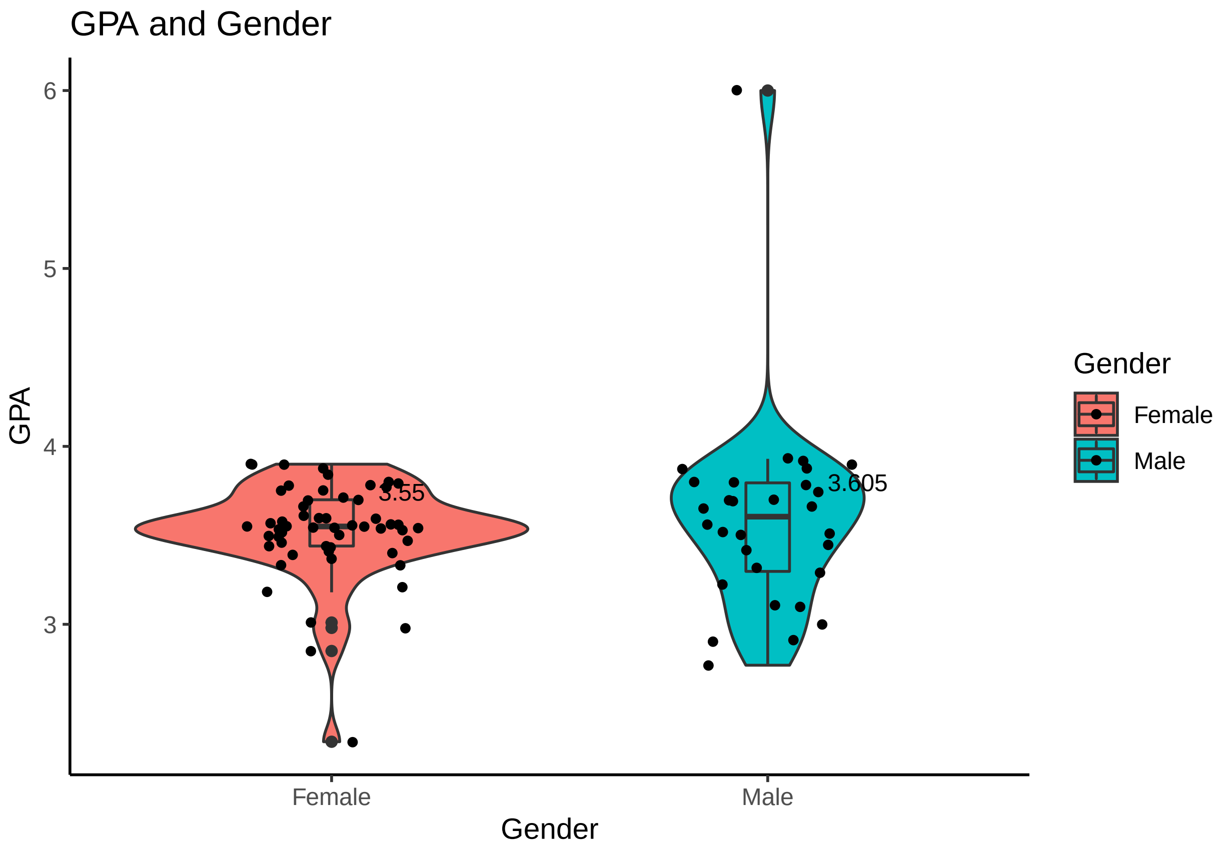
```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.5.3
```

```
## Create a table using the dataset
## This table is BY Gender
## The method is summarize
## A new column is med and is the median for GPA
(TEMPmeds <- ddply(MyData, .(Gender), summarize,
                   med = median(GPA)))
```

```
##   Gender   med
## 1 Female 3.550
## 2   Male 3.605
```

```
## Next, we have an incorrect value....let's SEE IT

(MyV1 <- ggplot(MyData, aes(x=Gender, y=GPA, fill=Gender)) +
    geom_violin(trim=TRUE)+ geom_boxplot(width=0.1)+
    geom_text(data = TEMPmeds,
              aes(x = Gender, y = med, label = med),
              size = 3, vjust = -1.5,hjust=-1)+
    ggtitle("GPA and Gender")+
    geom_jitter(shape=16, position=position_jitter(0.2)))
```



```
## Now we can SEE the issue. There is at least one GPA
## that is out of range. Let's fix this.
## Let's replace the missing GPA by finding the median
## for the ADMITS in that Gender group

## FIND the row with GPA > 4
(WrongGPAs <- MyData[(MyData$GPA<0 | MyData$GPA >4),])
```

```
##    Decision Gender    DateSub    State GPA WorkExp TestScore WritingScore
## 72    Admit   Male 2019-12-24 Colorado   6     0.8       969           93
##    VolunteerLevel
## 72              1
```

```
##
## We have Male Admit with a GPA of 6.

## Fix it by using Male Admit GPA Median
(Temp<-MyData[MyData$Decision=="Admit" & MyData$Gender=="Male",])
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 58    Admit   Male 2020-01-25    Florida 3.50     0.7       965           91
## 59    Admit   Male 2019-11-10   Colorado 3.65     1.7       963           90
## 60    Admit   Male 2019-12-21    Florida 3.66     2.2       967           91
## 61    Admit   Male 2019-12-03 California 3.69     3.2       967           93
## 62    Admit   Male 2019-11-26 California 3.70     1.4       966           94
## 63    Admit   Male 2019-11-01    Florida 3.70     3.7       969           99
## 64    Admit   Male 2019-11-16   Colorado 3.78     1.2       966            1
## 65    Admit   Male 2019-12-20    Florida 3.80     1.4       969           97
## 66    Admit   Male 2019-11-27    Florida 3.80     1.7       968           91
## 67    Admit   Male 2019-11-19 California 3.87     1.7       966           97
## 68    Admit   Male 2019-10-19 California 3.88     1.5       967           95
## 69    Admit   Male 2019-09-13 California 3.90     6.7       962          100
## 70    Admit   Male 2019-10-03   Colorado 3.92     1.2       969           95
## 71    Admit   Male 2019-11-02    Florida 3.93     0.8       969           99
## 72    Admit   Male 2019-12-24   Colorado 6.00     0.8       969           93
##    VolunteerLevel
## 58              1
## 59              1
## 60              2
## 61              3
## 62              0
## 63              2
## 64              4
## 65              4
## 66              3
## 67              5
## 68              5
## 69              0
## 70              3
## 71              4
## 72              1
```

```
## The median for Male Admits is:
(MyMed<-median(Temp$GPA, na.rm=TRUE))
```

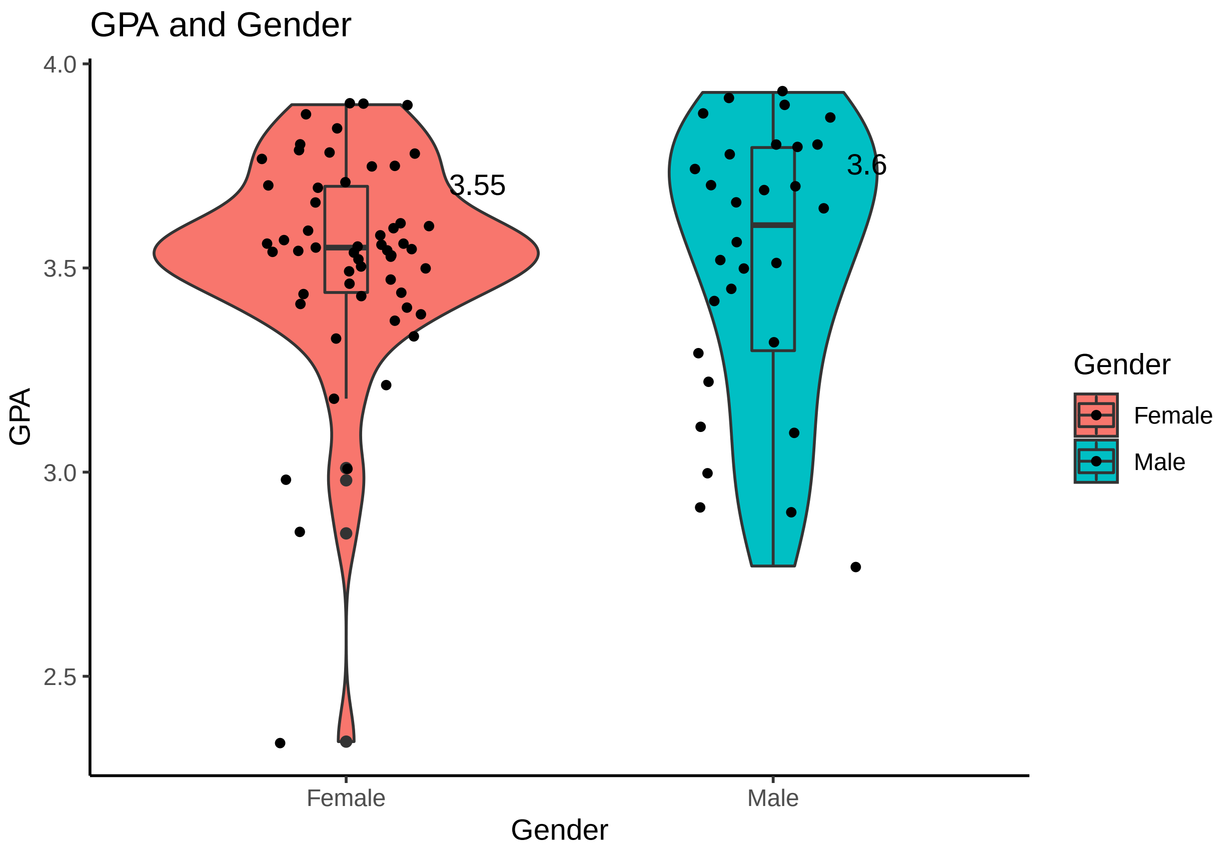
```
## [1] 3.8
```

```
## NOW - replace the missing GPA with this Median
MyData$GPA[MyData$GPA>4] <- MyMed

## NOW VISUALIZAE IT AGAIN:
(TEMPmeds <- ddply(MyData, .(Gender), summarize,
                   med = round(median(GPA),2)))
```

```
##   Gender  med
## 1 Female 3.55
## 2   Male 3.60
```

Fix it!!

```
(MyV1 <- ggplot(MyData, aes(x=Gender, y=GPA, fill=Gender)) +
    geom_violin(trim=TRUE)+ geom_boxplot(width=0.1)+
    geom_text(data = TEMPmeds,
              aes(x = Gender, y = med, label = med),
              size = 4, vjust = -2.5,hjust=-1.8)+
    ggtitle("GPA and Gender")+
    geom_jitter(shape=16, position=position_jitter(0.2)))
```



```
## That's better!

table(MyData$GPA)
```

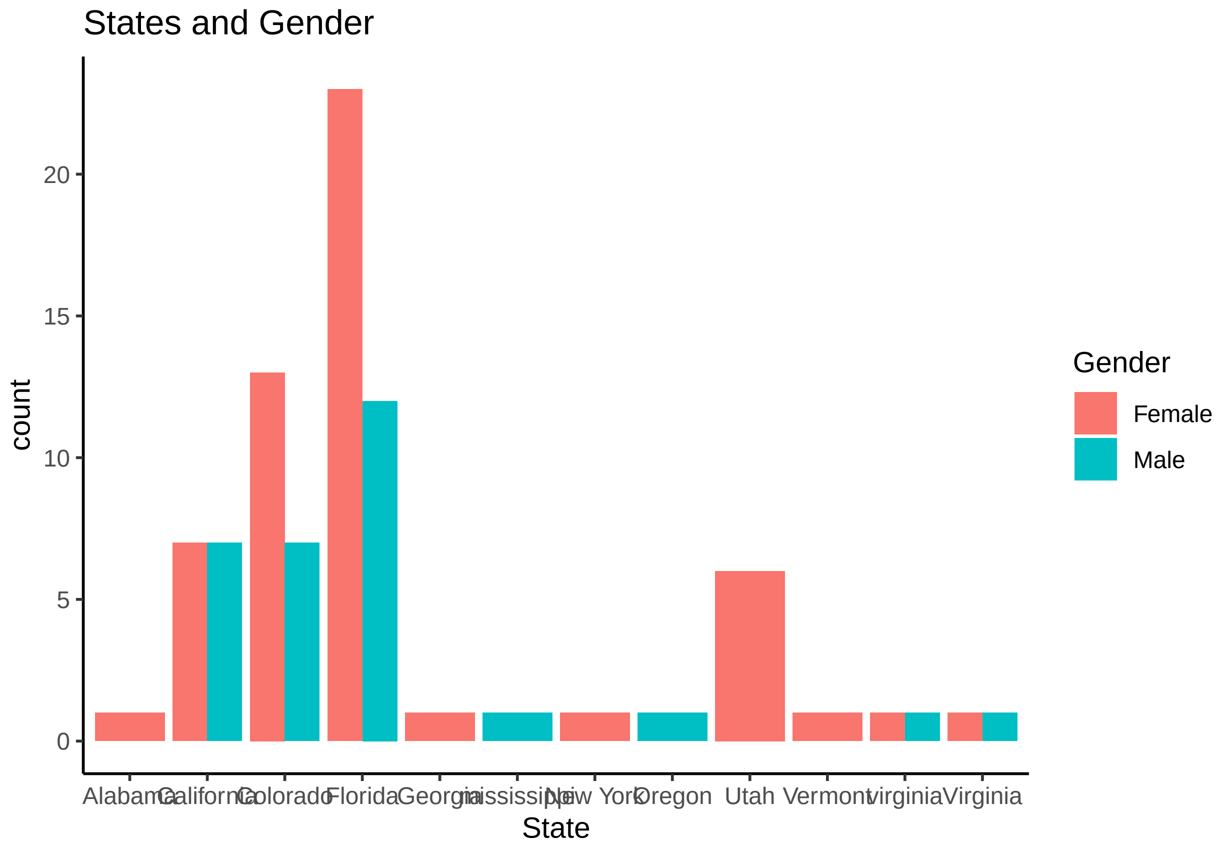
```
##
## 2.34 2.77 2.85  2.9 2.91 2.98    3 3.01  3.1 3.11 3.18 3.21 3.22 3.29 3.32 3.33
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2
## 3.37 3.39  3.4 3.41 3.42 3.43 3.44 3.45 3.46 3.47 3.49  3.5 3.51 3.52 3.53 3.54
##    1    1    1    1    1    1    2    1    1    1    1    3    1    2    2    4
## 3.55 3.56 3.57 3.58 3.59  3.6 3.61 3.65 3.66 3.69  3.7 3.71 3.74 3.75 3.77 3.78
##    3    4    1    1    1    2    1    1    2    1    4    1    1    2    1    3
## 3.79  3.8 3.84 3.87 3.88  3.9 3.92 3.93
##    1    4    1    1    2    4    1    1
## LOOKS GOOD!
```

State is next

```
#############################################
##
##             Let's look at State next
############################################
#names(MyData)
str(MyData$State)
```

```
##  Factor w/ 12 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
```

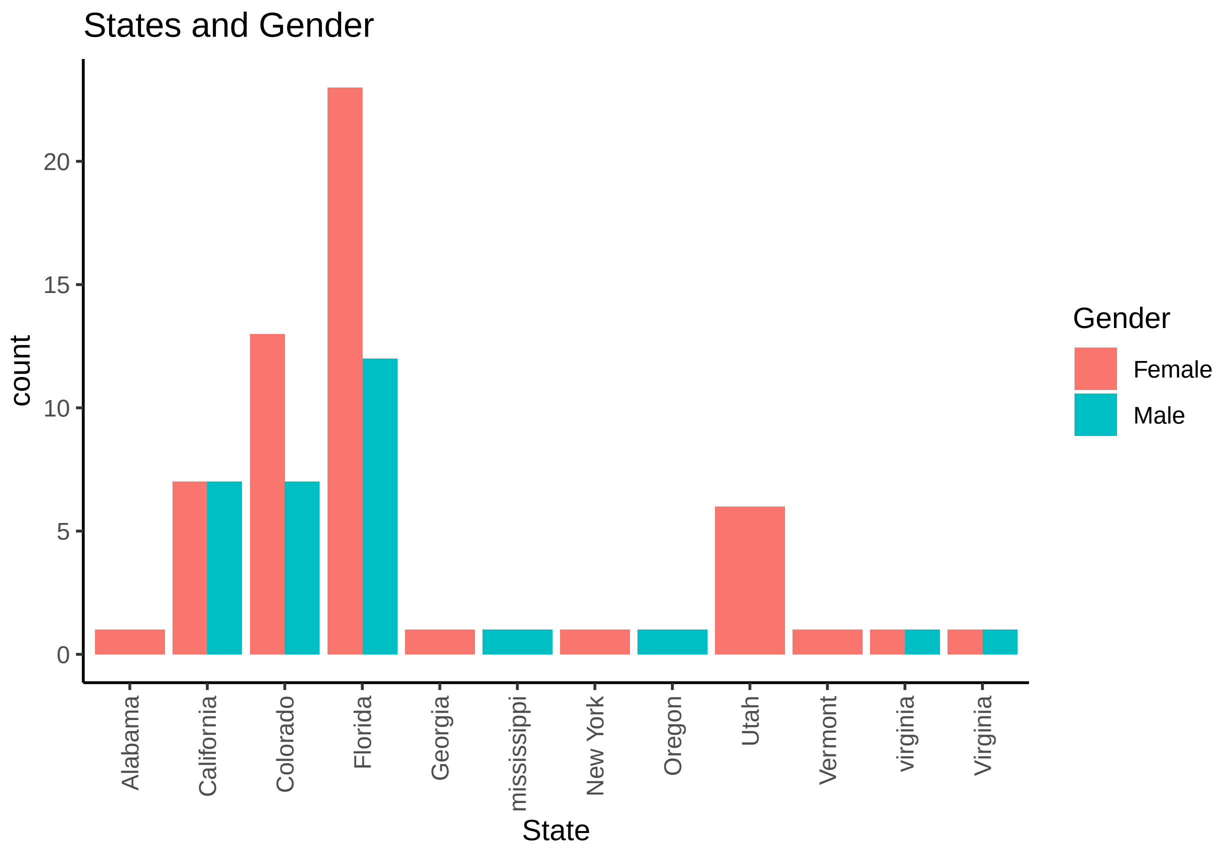
```
## Let's use a BAR to look
BaseGraph <- ggplot(MyData)
(MyG3<-BaseGraph +
    geom_bar(aes(State, fill = Gender), position="dodge")+
    ggtitle("States and Gender"))
```



```
## UGLY!!
```

This graph is not very aethestically pleasing . . . lets clean it up using “theme”s.

```r
## Let's make this nicer so we can READ THE X AXIS
(MyG3<-BaseGraph +
    geom_bar(aes(State, fill = Gender), position="dodge")+
    ggtitle("States and Gender")+
    theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)))
```



```r
## MUCH BETTER!
```

Now we can SEE that we have problems :(First, we have poor balance. It might be needed to collect all the lower count states, such as ALabama, Mississippi, etc. into a group called OTHER. However, we will not do this here. If you want to see how - look at this other tutorial http://drgates.georgetown.domains/SummerClassificationRMarkdown.html

Also - We have two Virginias (really!?!) - we need to combine them:

```r
MyData$State[MyData$State == "virginia"] <- "Virginia"
table(MyData$State)
```

```
##
##     Alabama  California    Colorado     Florida     Georgia mississippi
##           1          14          20          35           1           1
##    New York      Oregon        Utah     Vermont    virginia    Virginia
##           1           1           6           1           0           4
## Now - we need to remove the level of virginia
MyData$State<-as.character(MyData$State)
table(MyData$State)
```

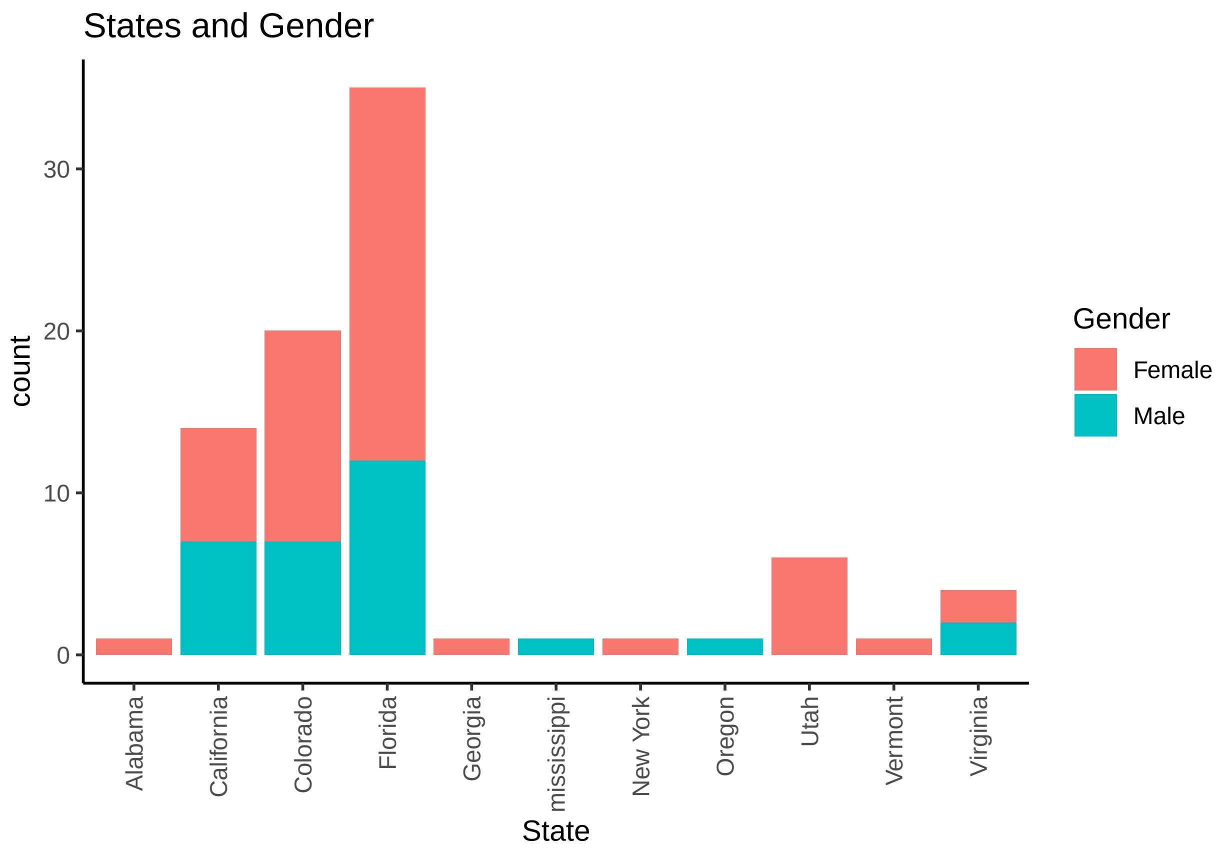
```
## 
##     Alabama  California    Colorado     Florida     Georgia mississippi 
##           1          14          20          35           1           1 
##    New York      Oregon        Utah     Vermont    Virginia 
##           1           1           6           1           4
```

```
MyData$State<-as.factor(MyData$State)
str(MyData$State)
```

```
##  Factor w/ 11 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
```

and confirm

```
## Check it
(MyG4<-ggplot(MyData) +
    geom_bar(aes(State, fill = Gender), position="stack")+
    ggtitle("States and Gender")+
    theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)))
```



Next: WorkExp

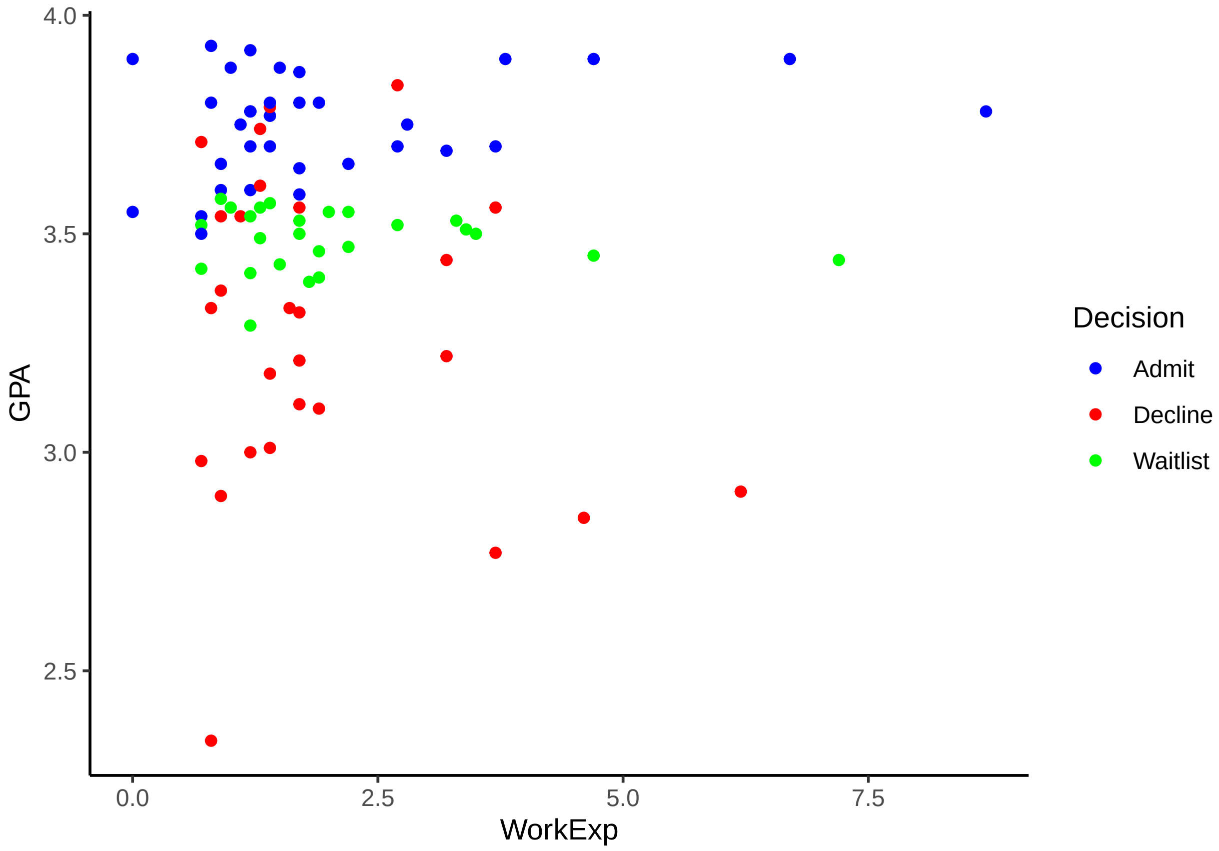
```
## Even better!

###########################################
##
## Now let's look at WorkExp
#########################################
#names(MyData)
(sum(is.na(MyData$WorkExp)))
```

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

```
str(MyData$WorkExp)
```

```
##  num [1:85] 0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
```

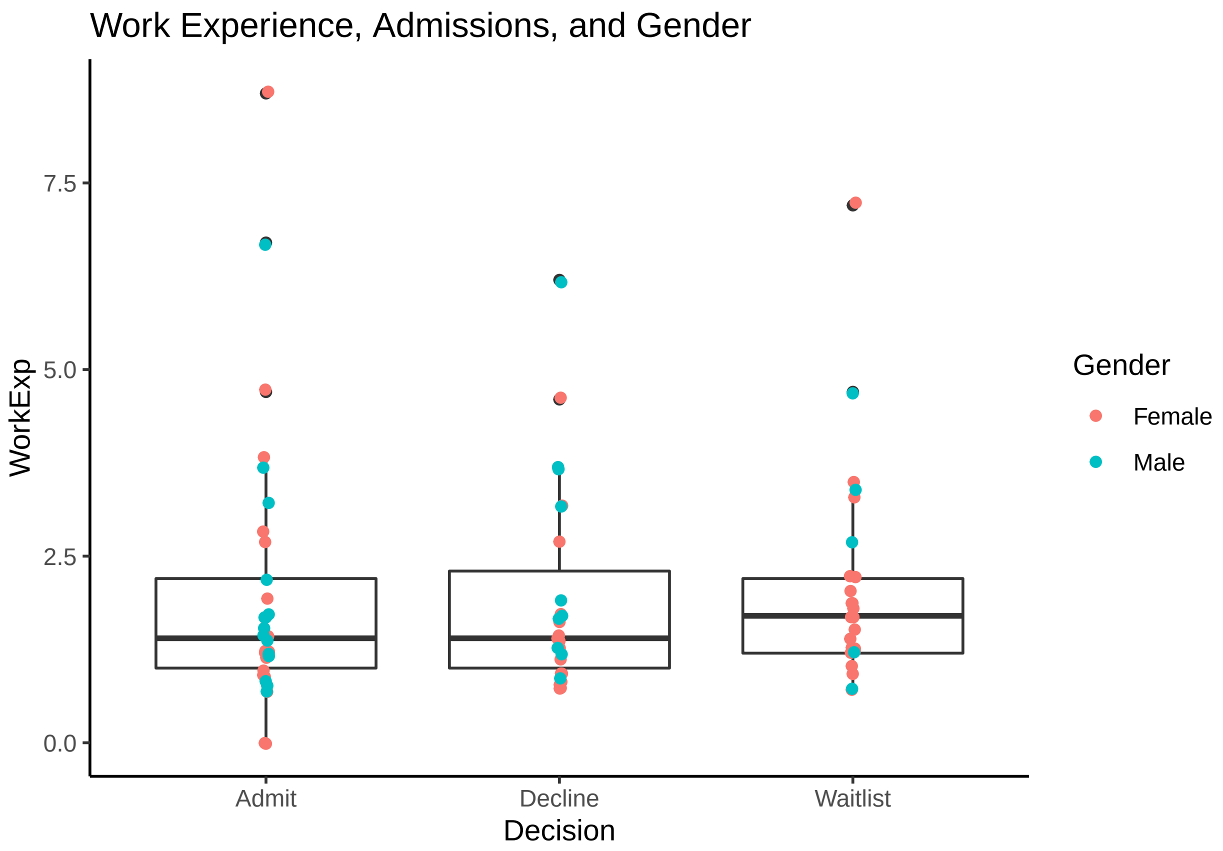
```
## Let's look
theme_set(theme_classic())

# Histogram on a Continuous (Numeric) Variable
(MyS3 <- ggplot(MyData,aes(x=WorkExp, y=GPA, color=Decision)) +
    geom_point() +
    scale_color_manual(values = c('blue',"red", "green")))
```



```
## This helps in many ways. We can see that we have no outliers
## or odd values.
```

However, let's check it with a box plot as well.

```
(MyL1<-ggplot(MyData, aes(x=Decision, y=WorkExp))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Work Experience, Admissions, and Gender"))
```



This looks good and it also starts to tell us that people were not penalized or prefered based on work experience.

Lets move on to TestScore and WritingScore.

```
#####################################################
##
##          Let's look at TestScore and Writing Score
##
#######################################################
(sum(is.na(MyData$TestScore)))
```

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

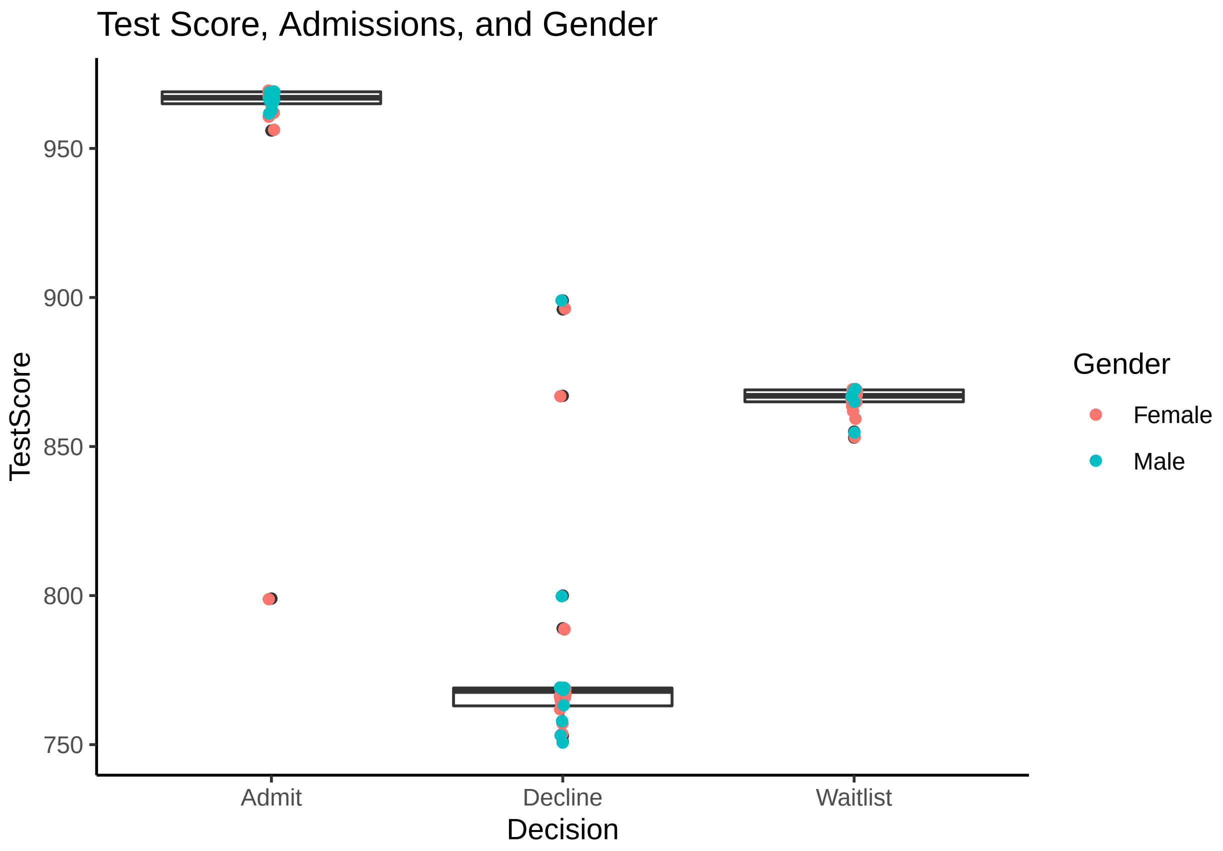
```
(sum(is.na(MyData$WritingScore)))
```

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

```
str(MyData)
```

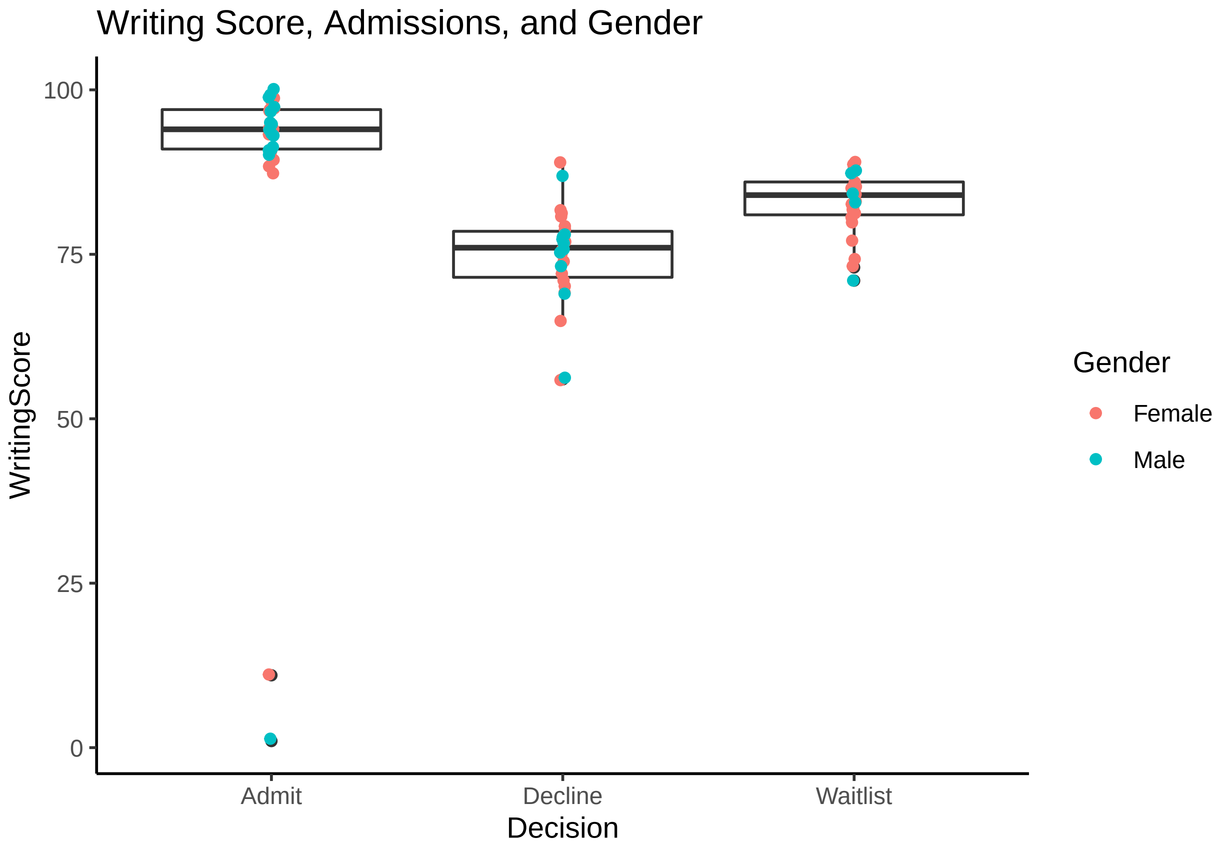
```
## 'data.frame':    85 obs. of  9 variables:
##  $ Decision      : Factor w/ 3 levels "Admit","Decline",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender        : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DateSub       : Date, format: "2020-01-11" "2020-01-11" ...
##  $ State         : Factor w/ 11 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
##  $ GPA           : num  3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
##  $ WorkExp       : num  0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
##  $ TestScore     : int  965 962 969 969 967 956 969 799 969 969 ...
##  $ WritingScore  : int  11 97 93 97 94 89 94 97 93 99 ...
##  $ VolunteerLevel: int  1 0 0 2 2 1 2 5 0 4 ...
```

```
## Box plots are great to look for odd values

(MyL2<-ggplot(MyData, aes(x=Decision, y=TestScore))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Test Score, Admissions, and Gender"))
```



Interesting!! This mostly makes sense except for the 800 in the Admit group. However, it is not an outlier - it is just interesting.

```
(MyL3<-ggplot(MyData, aes(x=Decision, y=WritingScore))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Writing Score, Admissions, and Gender"))
```



Hmmm - most of this looks OK, BUT, we have some very strangevalues for the Admit group. Let's look at these:

```
(Temp <- subset(MyData, Decision=="Admit",
               select=c(Decision,WritingScore)) )
```

```
##    Decision WritingScore
## 1     Admit           11
## 2     Admit           97
## 3     Admit           93
## 4     Admit           97
## 5     Admit           94
## 6     Admit           89
## 7     Admit           94
## 8     Admit           97
## 9     Admit           93
## 10    Admit           99
## 11    Admit           91
## 12    Admit           87
## 13    Admit           94
## 14    Admit           93
## 15    Admit           93
## 16    Admit           98
## 17    Admit           88
## 18    Admit           95
## 58    Admit           91
## 59    Admit           90
## 60    Admit           91
## 61    Admit           93
## 62    Admit           94
## 63    Admit           99
## 64    Admit            1
## 65    Admit           97
## 66    Admit           91
## 67    Admit           97
## 68    Admit           95
## 69    Admit          100
## 70    Admit           95
## 71    Admit           99
## 72    Admit           93
```

```
table(Temp$WritingScore)
```

```
## 
##   1  11  87  88  89  90  91  93  94  95  97  98  99 100 
##   1   1   1   1   1   1   4   6   4   3   5   1   3   1
```

OK - we can see that two score seem incorrect. The 1 and the 11, for an Admit, it not likely. Let's replace them with median

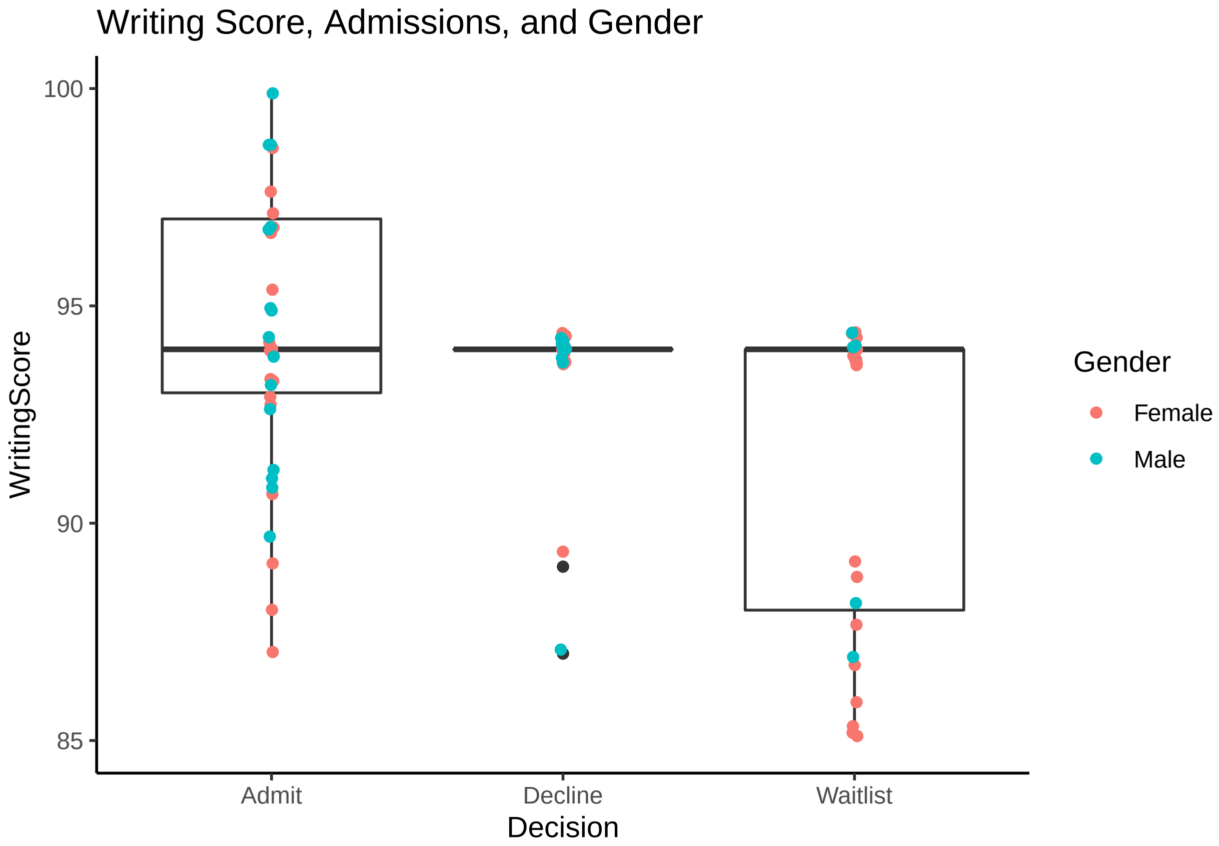
```
(Temp3<-MyData[MyData$Decision=="Admit",])
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1     Admit Female 2020-01-11    Florida 3.54     0.7       965           11
## 2     Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3     Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4     Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5     Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6     Admit Female 2019-11-03 California 3.66     0.9       956           89
## 7     Admit Female 2019-11-08 California 3.70     1.2       969           94
## 8     Admit Female 2019-10-07 California 3.70     2.7       799           97
## 9     Admit Female 2019-10-10   Colorado 3.75     1.1       969           93
## 10    Admit Female 2020-01-15    Florida 3.77     1.4       969           99
## 11    Admit Female 2019-10-31 California 3.78     8.7       966           91
## 12    Admit Female 2019-10-30       Utah 3.78     1.2       968           87
## 13    Admit Female 2019-10-14    Florida 3.80     1.9       965           94
## 14    Admit Female 2019-11-04   Colorado 3.88     1.0       969           93
## 15    Admit Female 2019-12-20    Florida 3.90     4.7       961           93
## 16    Admit Female 2019-10-25   Colorado 3.90     3.8       967           98
## 17    Admit Female 2019-12-28    Florida 3.90     0.0       967           88
## 18    Admit Female 2020-01-10 California 3.75     2.8       967            95
## 58    Admit   Male 2020-01-25    Florida 3.50     0.7       965            91
## 59    Admit   Male 2019-11-10   Colorado 3.65     1.7       963            90
## 60    Admit   Male 2019-12-21    Florida 3.66     2.2       967            91
## 61    Admit   Male 2019-12-03 California 3.69     3.2       967            93
## 62    Admit   Male 2019-11-26 California 3.70     1.4       966            94
## 63    Admit   Male 2019-11-01    Florida 3.70     3.7       969            99
## 64    Admit   Male 2019-11-16   Colorado 3.78     1.2       966             1
## 65    Admit   Male 2019-12-20    Florida 3.80     1.4       969            97
## 66    Admit   Male 2019-11-27    Florida 3.80     1.7       968            91
## 67    Admit   Male 2019-11-19 California 3.87     1.7       966            97
## 68    Admit   Male 2019-10-19 California 3.88     1.5       967            95
## 69    Admit   Male 2019-09-13 California 3.90     6.7       962           100
## 70    Admit   Male 2019-10-03   Colorado 3.92     1.2       969            95
## 71    Admit   Male 2019-11-02    Florida 3.93     0.8       969            99
## 72    Admit   Male 2019-12-24   Colorado 3.80     0.8       969            93
##    VolunteerLevel
## 1               1
## 2               0
## 3               0
## 4               2
## 5               2
## 6               1
## 7               2
## 8               5
## 9               0
## 10              4
## 11              2
## 12              2
## 13              5
## 14              4
## 15              1
## 16              3
## 17              0
## 18              3
## 58              1
## 59              1
## 60              2
## 61              3
## 62              0
## 63              2
## 64              4
## 65              4
## 66              3
## 67              5
## 68              5
## 69              0
## 70              3
## 71              4
## 72              1
```

```
## The median for Admits is:
(MyMed2<-median(Temp3$WritingScore, na.rm=TRUE))
```

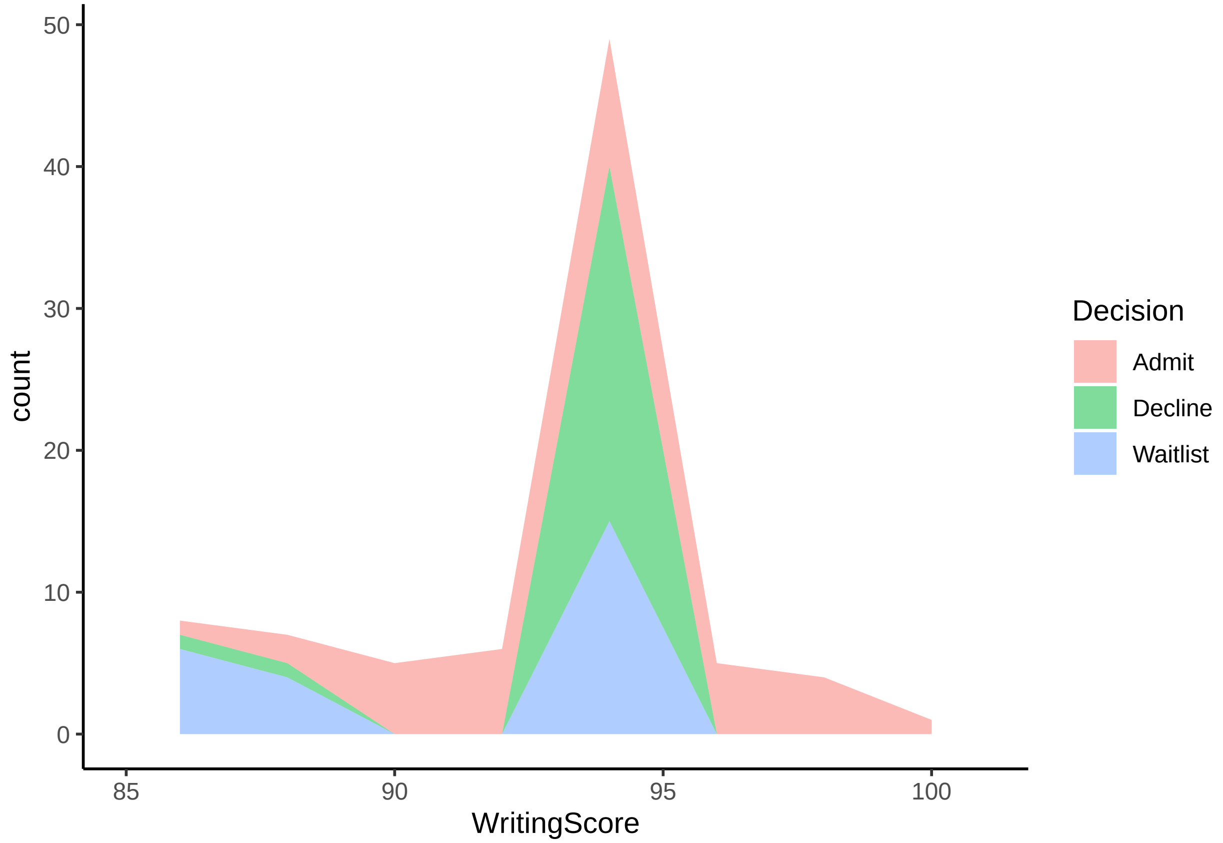
```
## [1] 94
```

```
## NOW - replace the incorrect  with this Median
MyData$WritingScore[MyData$WritingScore<85] <- MyMed2


## check again
(MyL4<-ggplot(MyData, aes(x=Decision, y=WritingScore))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Writing Score, Admissions, and Gender"))
```

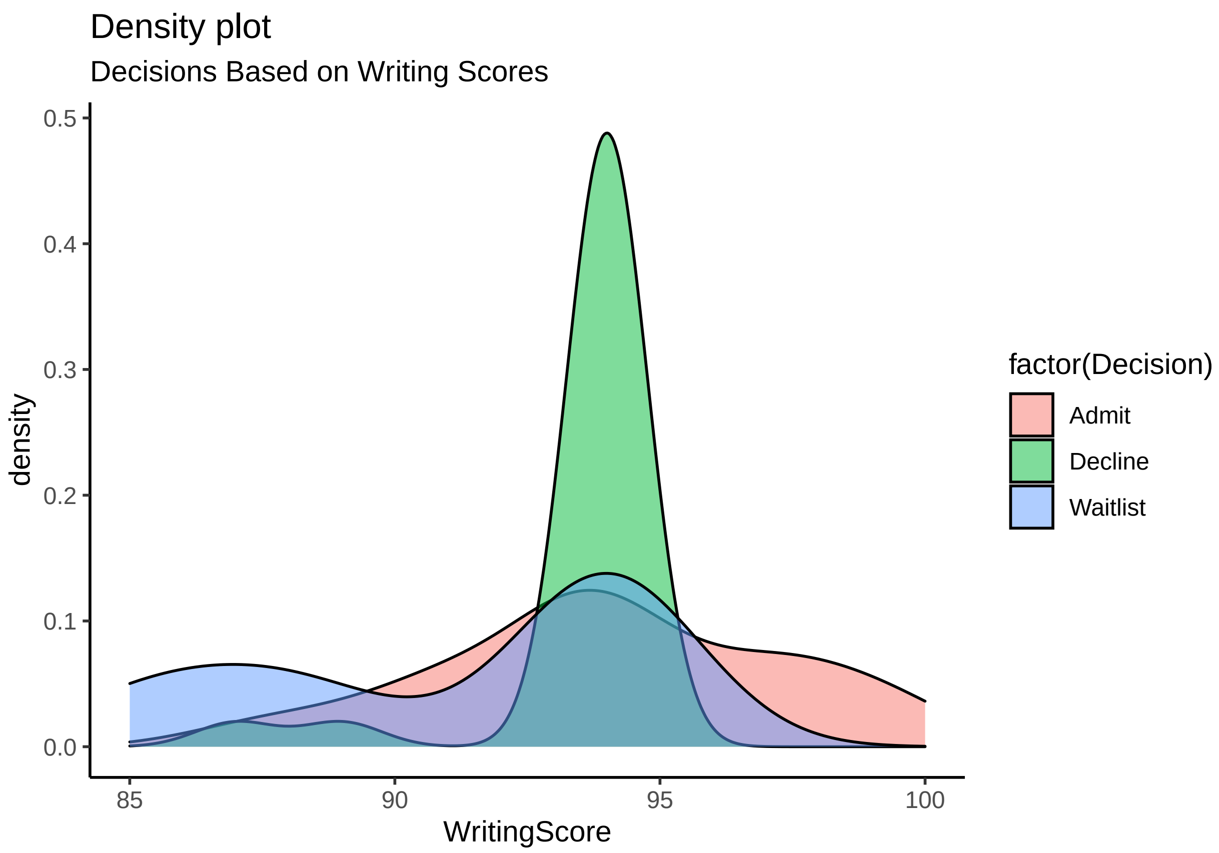


MUCH BETTER! We can also look using density area plots…

```
# Use semi-transparent fill
(MyPlot4<-ggplot(MyData, aes(x=WritingScore, fill=Decision)) +
    geom_area(stat ="bin", binwidth=2, alpha=0.5) +
    theme_classic())
```



```
## Here - using density - we can get a deeper look
MyPlot5 <- ggplot(MyData, aes(WritingScore))
MyPlot5 + geom_density(aes(fill=factor(Decision)), alpha=0.5) +
  labs(title="Density plot",
       subtitle="Decisions Based on Writing Scores")
```



EDA

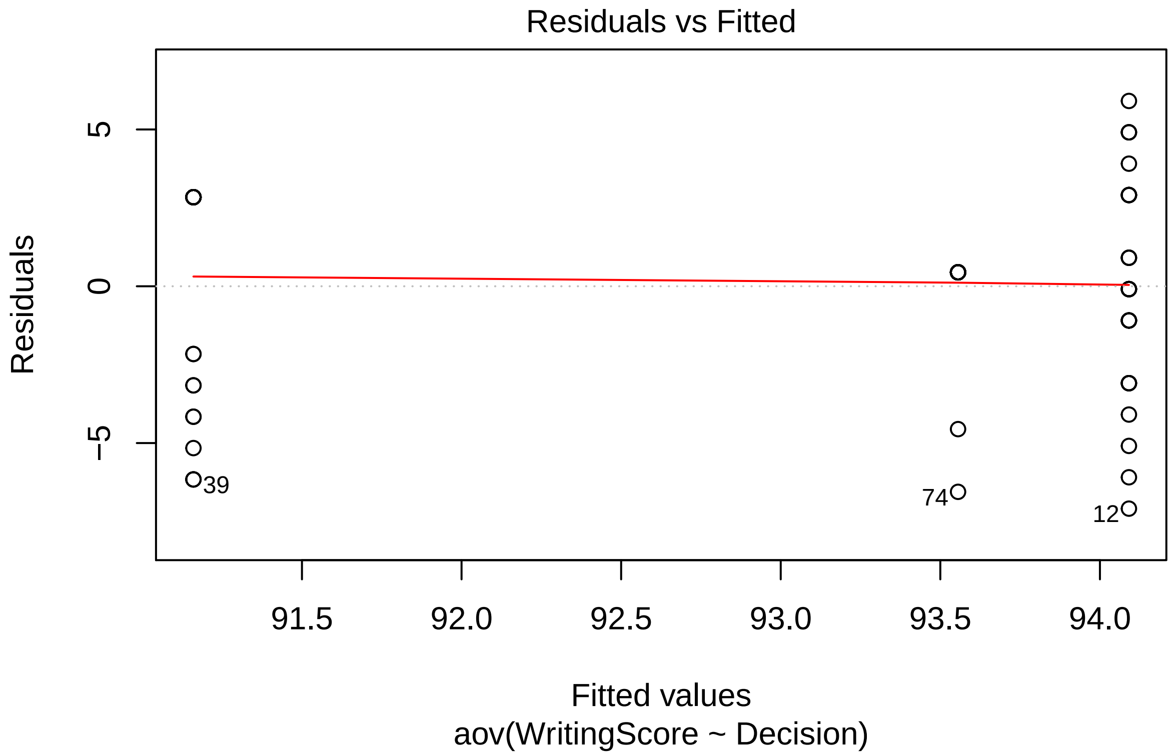
Let investigate some of these variables for associations with our dependent variable – EDA. Remember our goal is to leverage this data for prediction, decision-making, etc.

Does it seem like WritingScore is really related to Admissions?

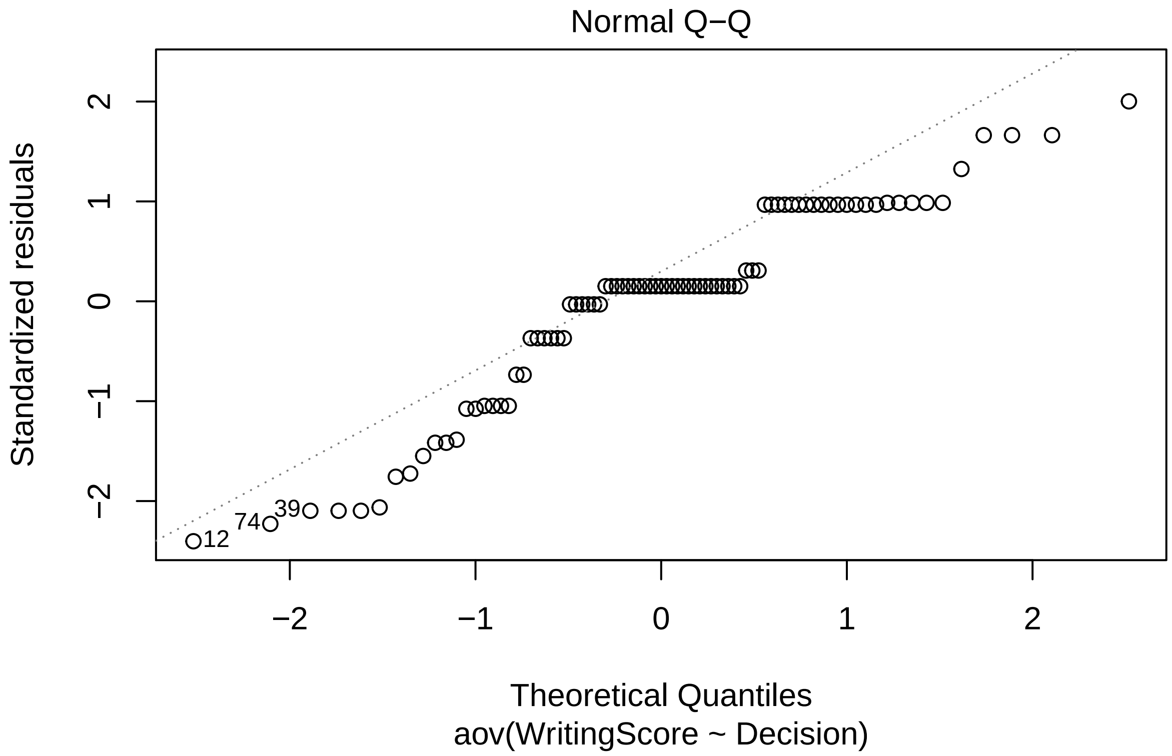
```
## Let's run an ANOVA test to see
MyANOVA_WS_Adm <- aov(WritingScore ~ Decision, data = MyData)
# Summary of the analysis
summary(MyANOVA_WS_Adm)  ## The test IS significant!
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Decision     2  132.0   65.98   7.343 0.00117 **
## Residuals   82  736.8    8.98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(MyANOVA_WS_Adm, 1)
```



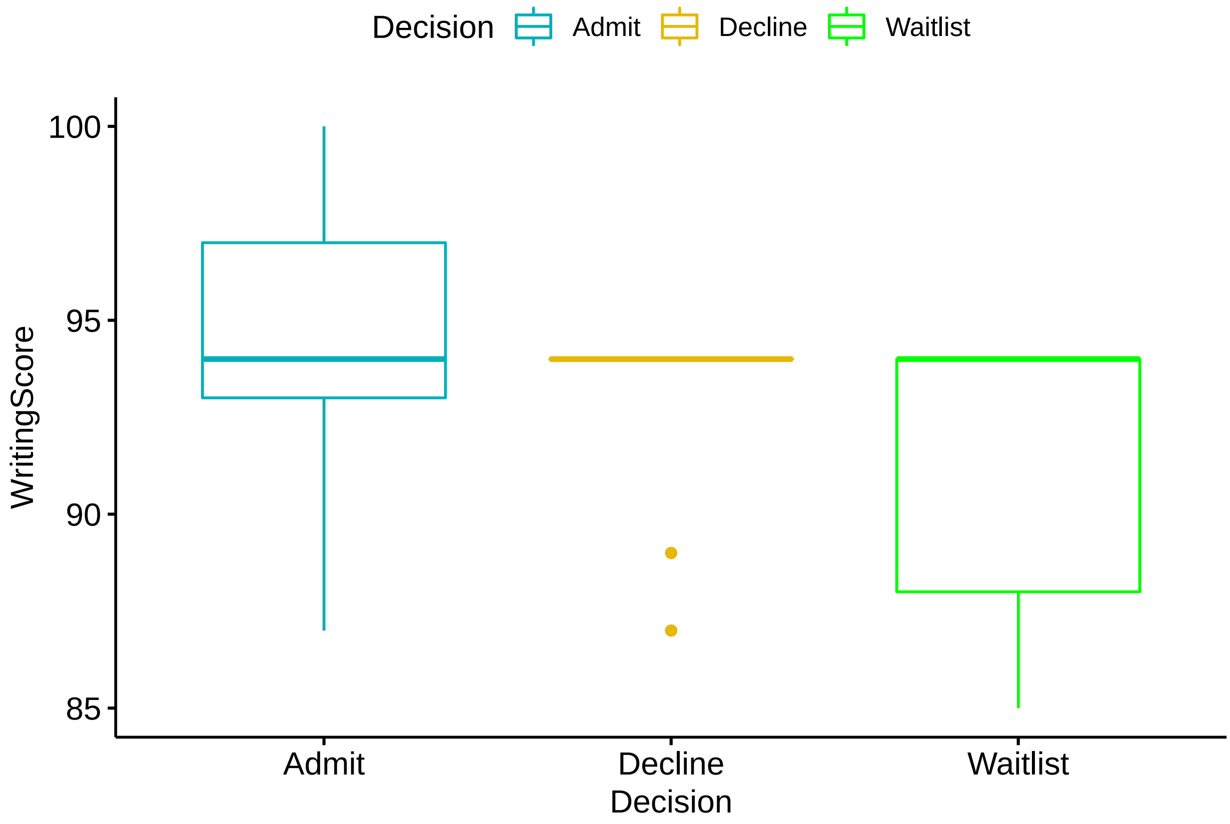
```
## The above shows we can assume the homogeneity of variances.
plot(MyANOVA_WS_Adm, 2) ## Close to normal
```



```
library("ggpubr")
```

```
##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':
##
##     mutate
```

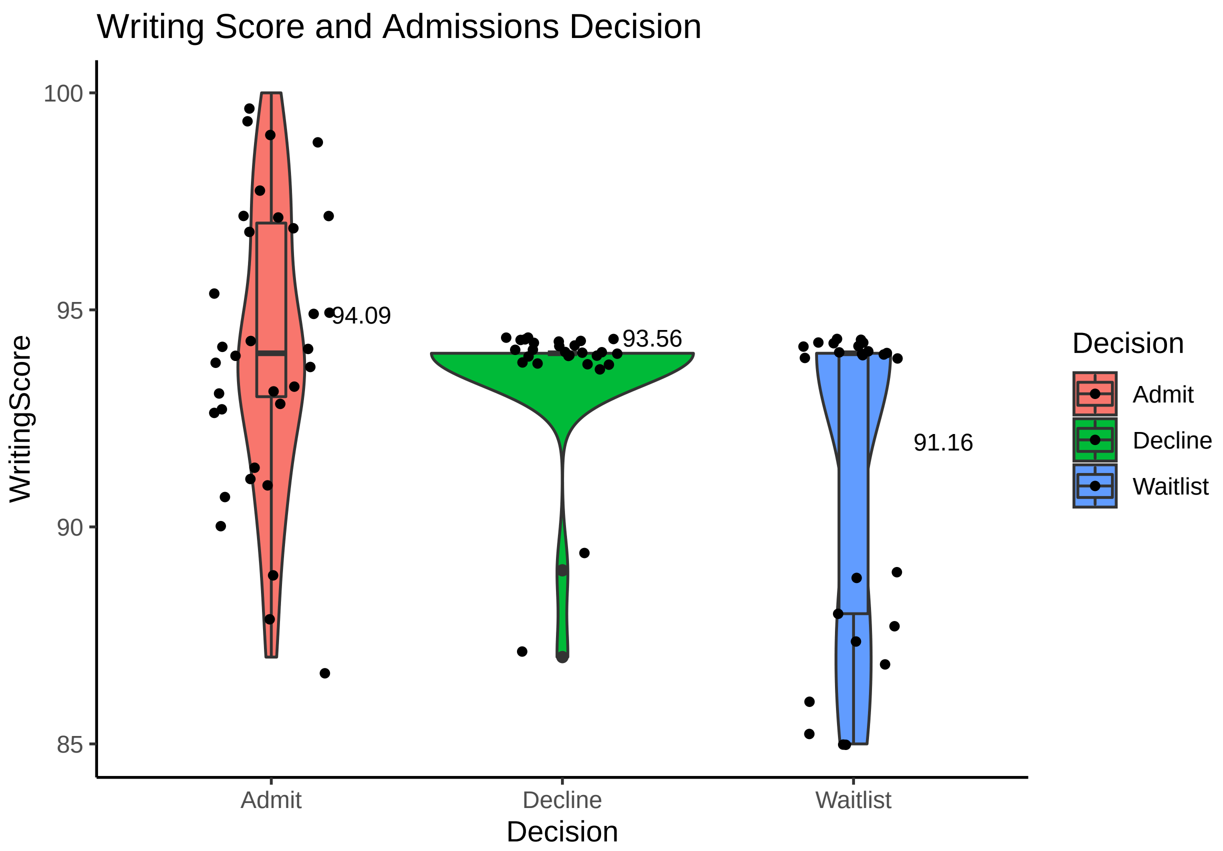
```
ggboxplot(MyData, x = "Decision", y = "WritingScore",
          color = "Decision", palette = c("#00AFBB", "#E7B800","green"),
          ylab = "WritingScore", xlab = "Decision")
```



```
## Let's add labels...

(TheMean <- ddply(MyData, .(Decision), summarize,
                  mean2 = round(  mean(WritingScore) ,2 )))
```

```
##   Decision mean2
## 1    Admit 94.09
## 2  Decline 93.56
## 3 Waitlist 91.16
```

```
## Another View...

(MyV2 <- ggplot(MyData, aes(x=Decision, y=WritingScore, fill=Decision)) +
    geom_violin(trim=TRUE)+ geom_boxplot(width=0.1)+
    geom_text(data = TheMean,
              aes(x = Decision, y = mean2, label = mean2),
              size = 3, vjust = -1.5,hjust=-1)+
    ggtitle("Writing Score and Admissions Decision")+
    geom_jitter(shape=16, position=position_jitter(0.2)))
```



And lastly … VolunteerLevel

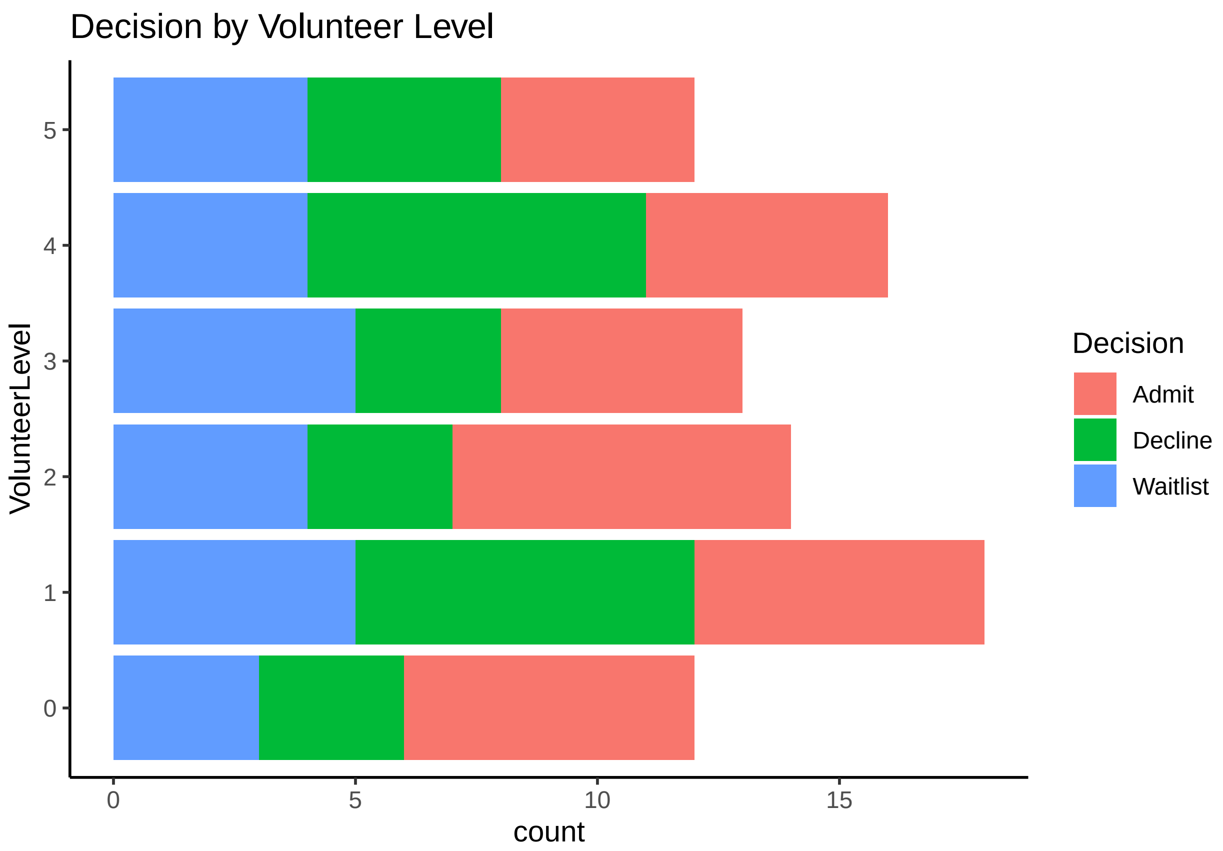
```
############################################
##  The last variable is VolunteerLevel
##
###############################################
str(MyData$VolunteerLevel)
```

```
##  int [1:85] 1 0 0 2 2 1 2 5 0 4 ...
```

```
## This should NOT be an int
## COrrect it to factor
MyData$VolunteerLevel <- as.factor(MyData$VolunteerLevel)
table(MyData$VolunteerLevel)
```

```
##
##  0  1  2  3  4  5
## 12 18 14 13 16 12
```

```
(MyG1<-ggplot(MyData) +
    geom_bar(aes(VolunteerLevel, fill = Decision)) +
    ggtitle("Decision by Volunteer Level")+
    coord_flip())
```



This is a good starting point for some more extended EDA. Note that the first steps were to load and clean the data. We can then confirm the tidy-ness of the data visually. Next it is time to INVESTIGATE the data – EDA. We try to answer the question, how can we best leverage the data. If our research problem or goals was attempting to predict admissions based on these variables, we should assess the associations / correlations of these variables with our admissions variable (as we did in some instances above.)

This is a really good starting point for some more investigation, exploration and visualization that would be incorporated into a comprehensive EDA.

```
################################################

##################################################
## Machine Learning Methods and Visualization
##
## Clustering
## Association Rule Mining
##
## Naive Bayes
## Decision Trees
## SVMs
##
##################################################

## PART 1: Unsupervised - Clustering
##         k-means and hierarchical

##-----------------------------------------------
```

```
#################################
#
#           HUGELY IMPORTANT
#
####### FOr any model or method
####### Make sure the data is
####### in the right (and a smart)
####### format. For example - for
####### clustering - you must REMOVE
####### the labels!
#
#####################################

MyData
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1     Admit Female 2020-01-11    Florida 3.54     0.7       965           94
## 2     Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3     Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4     Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5     Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6     Admit Female 2019-11-03 California 3.66     0.9       956           89
## 7     Admit Female 2019-11-08 California 3.70     1.2       969           94
## 8     Admit Female 2019-10-07 California 3.70     2.7       799           97
## 9     Admit Female 2019-10-10   Colorado 3.75     1.1       969           93
## 10    Admit Female 2020-01-15    Florida 3.77     1.4       969           99
## 11    Admit Female 2019-10-31 California 3.78     8.7       966           91
## 12    Admit Female 2019-10-30       Utah 3.78     1.2       968           87
## 13    Admit Female 2019-10-14    Florida 3.80     1.9       965           94
## 14    Admit Female 2019-11-04   Colorado 3.88     1.0       969           93
## 15    Admit Female 2019-12-20    Florida 3.90     4.7       961           93
## 16    Admit Female 2019-10-25   Colorado 3.90     3.8       967           98
## 17    Admit Female 2019-12-28    Florida 3.90     0.0       967           88
## 18    Admit Female 2020-01-10 California 3.75     2.8       967           95
## 20  Decline Female 2020-01-14 California 2.34     0.8       754           94
## 21  Decline Female 2020-01-31   Colorado 2.85     4.6       762           94
## 22  Decline Female 2020-01-10   Colorado 2.98     0.7       763           94
## 23  Decline Female 2020-01-25       Utah 3.01     1.4       769           94
## 24  Decline Female 2020-02-27    Florida 3.18     1.4       768           94
## 25  Decline Female 2019-12-31   Virginia 3.21     1.7       766           94
## 26  Decline Female 2020-03-06    Florida 3.33     1.6       766           94
## 27  Decline Female 2020-02-07       Utah 3.33     0.8       768           94
## 28  Decline Female 2019-12-03   Virginia 3.37     0.9       766           94
## 29  Decline Female 2019-11-30   Colorado 3.44     3.2       757           94
## 30  Decline Female 2020-01-12    Florida 3.54     0.9       765           94
## 31  Decline Female 2020-02-15    Florida 3.54     1.1       767           94
## 32  Decline Female 2019-12-10    Florida 3.56     1.7       769           94
## 33  Decline Female 2020-01-22    Florida 3.61     1.3       789           94
## 34  Decline Female 2019-12-04    Florida 3.71     0.7       789           94
## 35  Decline Female 2019-11-25    Florida 3.79     1.4       867           94
## 36  Decline Female 2020-01-12    Florida 3.84     2.7       896           89
## 37 Waitlist Female 2019-12-30       Utah 3.39     1.8       866           94
## 38 Waitlist Female 2020-02-19    Florida 3.40     1.9       859           88
## 39 Waitlist Female 2019-12-09     Alabama 3.41     1.2       868           85
## 40 Waitlist Female 2019-12-23    Colorado 3.43     1.5       869           85
## 41 Waitlist Female 2020-01-29     Florida 3.44     7.2       865           94
## 42 Waitlist Female 2020-02-16     Florida 3.46     1.9       869           89
## 43 Waitlist Female 2020-01-17  California 3.47     2.2       867           94
## 44 Waitlist Female 2019-12-27    Colorado 3.49     1.3       866           86
## 45 Waitlist Female 2020-02-04     Florida 3.50     1.7       869           94
## 46 Waitlist Female 2019-12-04    Colorado 3.50     3.5       869           94
## 47 Waitlist Female 2020-01-23     Florida 3.52     0.7       868           94
## 48 Waitlist Female 2020-01-30     Florida 3.53     1.7       869           87
## 49 Waitlist Female 2019-11-28     Vermont 3.53     3.3       862           85
## 50 Waitlist Female 2019-11-04    Colorado 3.54     1.2       868           94
## 51 Waitlist Female 2019-12-08    New York 3.55     2.2       866           94
## 52 Waitlist Female 2019-12-11     Florida 3.55     2.0       853           94
## 53 Waitlist Female 2019-12-06     Georgia 3.56     1.0       866           89
## 54 Waitlist Female 2019-11-17     Florida 3.56     1.3       869           94
## 55 Waitlist Female 2019-11-15        Utah 3.57     1.4       869           94
## 56 Waitlist Female 2019-11-09        Utah 3.58     0.9       864           94
## 58    Admit   Male 2020-01-25     Florida 3.50     0.7       965           91
## 59    Admit   Male 2019-11-10    Colorado 3.65     1.7       963           90
## 60    Admit   Male 2019-12-21     Florida 3.66     2.2       967           91
## 61    Admit   Male 2019-12-03  California 3.69     3.2       967           93
## 62    Admit   Male 2019-11-26  California 3.70     1.4       966           94
## 63    Admit   Male 2019-11-01     Florida 3.70     3.7       969           99
## 64    Admit   Male 2019-11-16    Colorado 3.78     1.2       966           94
## 65    Admit   Male 2019-12-20     Florida 3.80     1.4       969           97
## 66    Admit   Male 2019-11-27     Florida 3.80     1.7       968           91
## 67    Admit   Male 2019-11-19  California 3.87     1.7       966           97
## 68    Admit   Male 2019-10-19  California 3.88     1.5       967           95
## 69    Admit   Male 2019-09-13  California 3.90     6.7       962          100
## 70    Admit   Male 2019-10-03    Colorado 3.92     1.2       969           95
## 71    Admit   Male 2019-11-02     Florida 3.93     0.8       969           99
## 72    Admit   Male 2019-12-24    Colorado 3.80     0.8       969           93
## 73  Decline   Male 2020-02-15    Virginia 2.77     3.7       763           94
## 74  Decline   Male 2020-02-01      Oregon 2.90     0.9       769           87
## 75  Decline   Male 2020-02-11 mississippi 2.91     6.2       753           94
## 76  Decline   Male 2020-01-15    Colorado 3.00     1.2       768           94
## 77  Decline   Male 2020-03-20     Florida 3.10     1.9       751           94
## 78  Decline   Male 2020-02-01    Colorado 3.11     1.7       758           94
## 79  Decline   Male 2020-01-05    Virginia 3.22     3.2       769           94
## 80  Decline   Male 2019-12-25    Colorado 3.32     1.7       768           94
## 81  Decline   Male 2020-01-05     Florida 3.56     3.7       899           94
## 82  Decline   Male 2019-12-30     Florida 3.74     1.3       800           94
## 83 Waitlist   Male 2020-01-28     Florida 3.29     1.2       869           94
## 84 Waitlist   Male 2019-12-01  California 3.42     0.7       869           94
## 85 Waitlist   Male 2020-02-10     Florida 3.45     4.7       867           94
## 86 Waitlist   Male 2019-12-29     Florida 3.51     3.4       865           88
## 87 Waitlist   Male 2019-11-07  California 3.52     2.7       855           87
##    VolunteerLevel
## 1               1
## 2               0
## 3               0
## 4               2
## 5               2
## 6                 1
## 7                 2
## 8                 5
## 9                 0
## 10                4
## 11                2
## 12                2
## 13                5
## 14                4
## 15                1
## 16                3
## 17                0
## 18                3
## 20                1
## 21                4
## 22                1
## 23                2
## 24                0
## 25                5
## 26                5
## 27                1
## 28                2
## 29                3
## 30                0
## 31                4
## 32                4
## 33                5
## 34                4
## 35                2
## 36                1
## 37                5
## 38                4
## 39                0
## 40                1
## 41                2
## 42                0
## 43                5
## 44                5
## 45                2
## 46                4
## 47                4
## 48                2
## 49                1
## 50                2
## 51                1
## 52                1
## 53                1
## 54                3
## 55                0
## 56                5
## 58                1
## 59                1
## 60                2
## 61                3
## 62              0
## 63              2
## 64              4
## 65              4
## 66              3
## 67              5
## 68              5
## 69              0
## 70              3
## 71              4
## 72              1
## 73              5
## 74              4
## 75              1
## 76              1
## 77              0
## 78              3
## 79              1
## 80              4
## 81              3
## 82              4
## 83              4
## 84              3
## 85              3
## 86              3
## 87              3
```

```
## To cluster - remove the label
## and remove any non-numeric variables

MyClusterData<-MyData[,c(5,6,7,8)]
head(MyClusterData)
```

```
##    GPA WorkExp TestScore WritingScore
## 1 3.54     0.7       965           94
## 2 3.55     0.0       962           97
## 3 3.59     1.7       969           93
## 4 3.60     0.9       969           97
## 5 3.60     1.2       967           94
## 6 3.66     0.9       956           89
```

```
## Next - normalize!
## This is especially important if you
## plan to use cosine sim as a distance
## measure.

## MIN - MAX Function
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

Norm_Data<-normalize(MyClusterData)
head(Norm_Data)
```

```
##           GPA      WorkExp TestScore WritingScore
## 1 0.003653251 0.0007223942 0.9958720   0.09700722
## 2 0.003663571 0.0000000000 0.9927761   0.10010320
## 3 0.003704850 0.0017543860 1.0000000   0.09597523
## 4 0.003715170 0.0009287926 1.0000000   0.10010320
## 5 0.003715170 0.0012383901 0.9979360   0.09700722
## 6 0.003777090 0.0009287926 0.9865841   0.09184727
```

```
##OK! How we have a dataframe that this
## appropriate for clustering

library(mclust)
```

```
## Warning: package 'mclust' was built under R version 3.5.3
```

```
## Package 'mclust' version 5.4.6
## Type 'citation("mclust")' for citing this R package in publications.
```

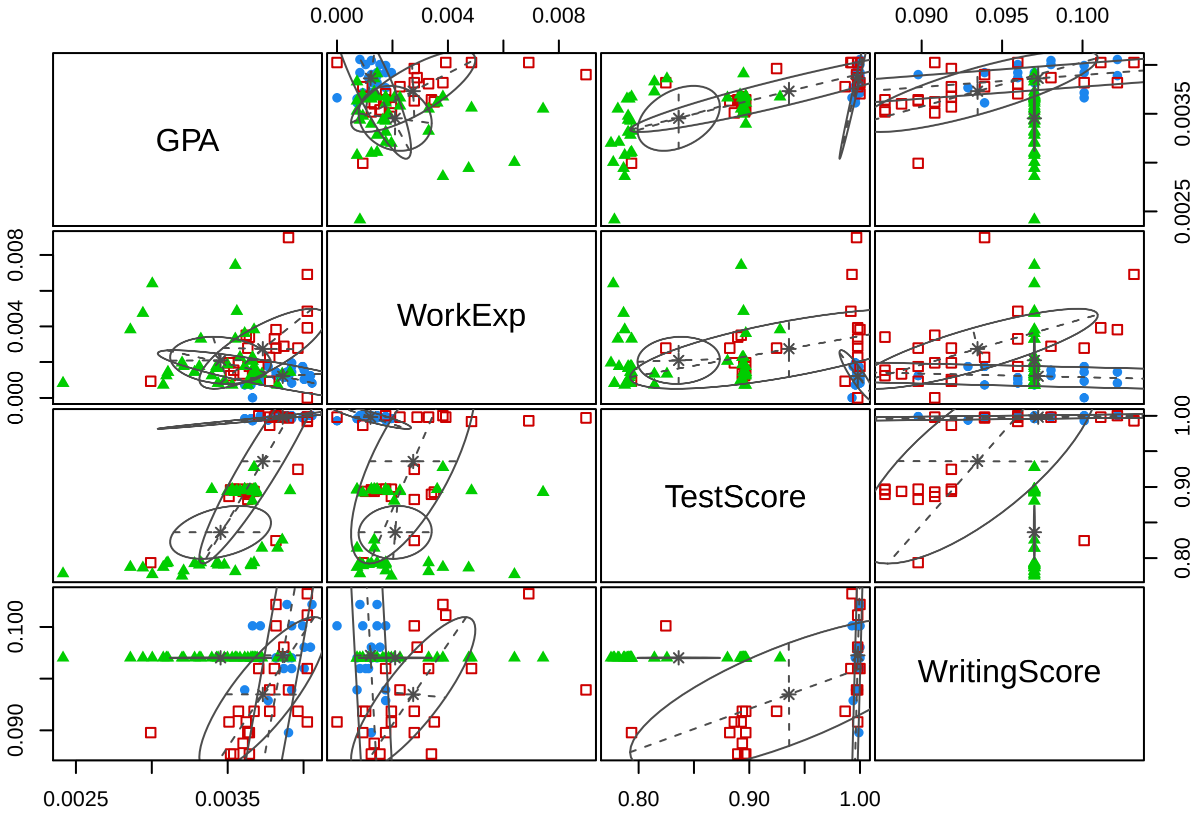
```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.5.3
```

```
library(cluster)

ClusFIT1 <- Mclust(Norm_Data,G=3)
summary(ClusFIT1)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 3 components:
##
##  log-likelihood  n df      BIC      ICL
##        1661.807 85 38 3154.794 3153.098
##
## Clustering table:
##  1  2  3
## 21 24 40
```

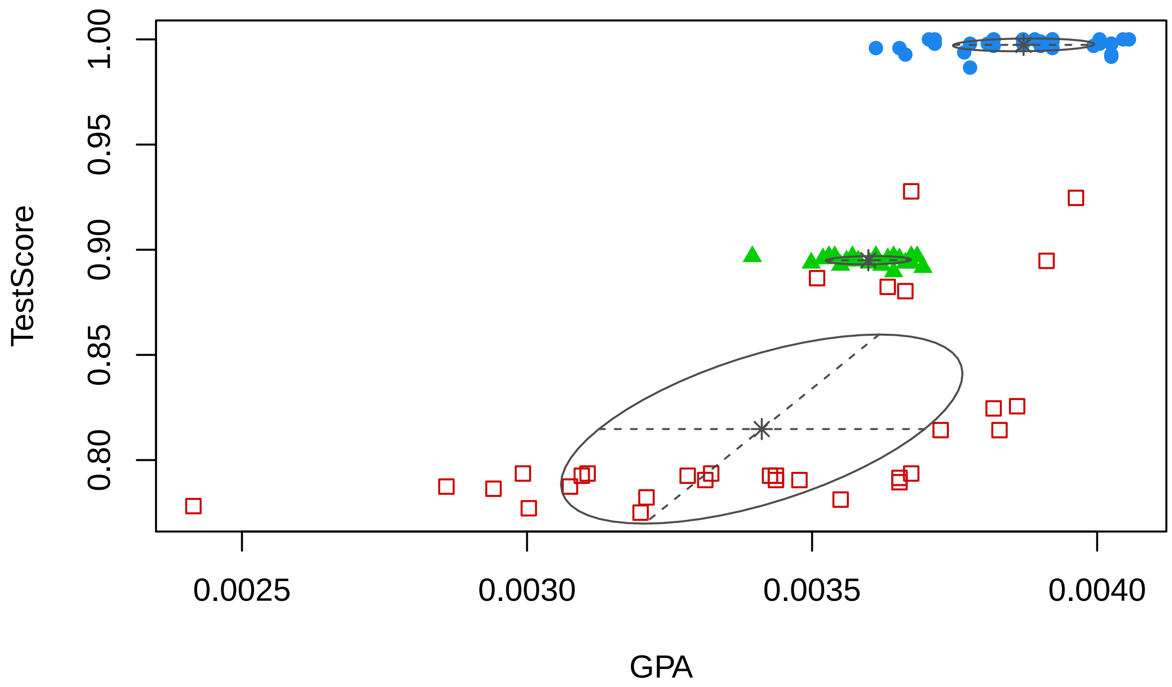
```
plot(ClusFIT1, what = "classification")
```



```
## Since we know that GPA and TestScore
## are most related to decision - let's
## LOOK at just those


ClusFIT2 <- Mclust(Norm_Data[,c(1,3)],G=3)
summary(ClusFIT2)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 3 components:
##
##  log-likelihood  n df      BIC      ICL
##        835.9479 85 15 1605.256 1604.166
##
## Clustering table:
##  1  2  3
## 32 31 22
```

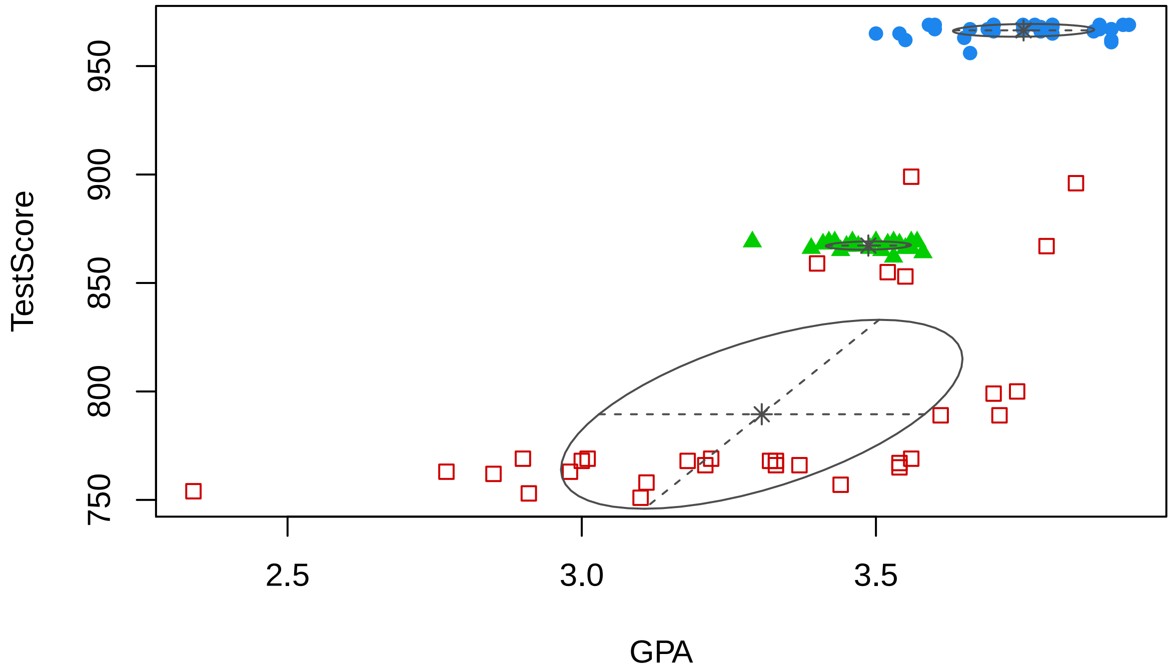
```
plot(ClusFIT2, what = "classification")
```



```
## Let's also look at this without
## normalized data

ClusFIT3 <- Mclust(MyClusterData[,c(1,3)],G=3)
summary(ClusFIT3)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 3 components:
##
##  log-likelihood  n df       BIC       ICL
##       -333.0171 85 15 -732.6739 -733.7645
##
## Clustering table:
##  1  2  3
## 32 31 22
```

```
plot(ClusFIT3, what = "classification")
```



```
## What does this show?
##
## Here, we SEE that we have clear clusters
## for Admit, Waitlist, and Decline.

################
## Example 2
C_Data<-MyClusterData[,c(3,1)]

## Add row names for the cluster vis
## Create "unique" but useful row names...

rownames(C_Data) <- paste((as.character(MyClusterData[,1])),
                    "_",
                    rownames(C_Data), sep="")

str(C_Data)
```

```
## 'data.frame':    85 obs. of  2 variables:
##  $ TestScore: int  965 962 969 969 967 956 969 799 969 969 ...
##  $ GPA      : num  3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
```

```
kmeansFIT <- kmeans(C_Data,3)
(kmeansFIT)
```

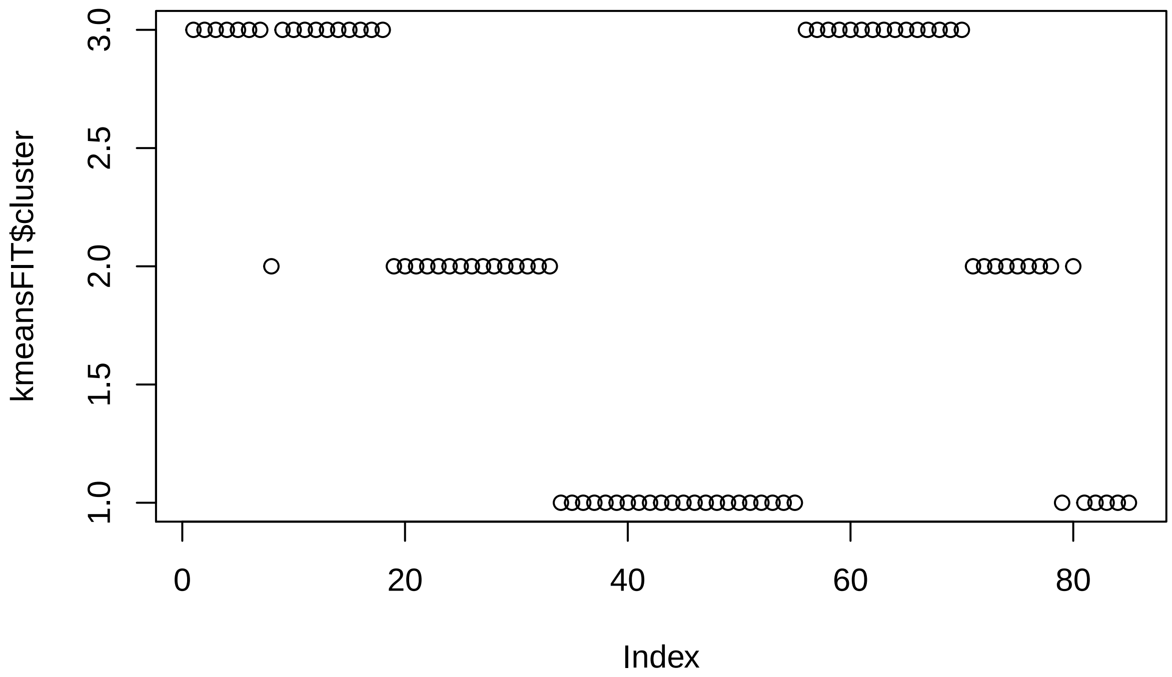
```
## K-means clustering with 3 clusters of sizes 28, 25, 32
```

```
##
## Cluster means:
##   TestScore      GPA
## 1  868.1429 3.512857
## 2  768.6400 3.230800
## 3  966.4688 3.750938
##
## Clustering vector:
##  3.54_1  3.55_2  3.59_3   3.6_4   3.6_5  3.66_6   3.7_7   3.7_8  3.75_9 3.77_10
##       3       3       3       3       3       3       3       2       3       3
## 3.78_11 3.78_12  3.8_13 3.88_14  3.9_15  3.9_16  3.9_17 3.75_18 2.34_20 2.85_21
##       3       3       3       3       3       3       3       3       2       2
## 2.98_22 3.01_23 3.18_24 3.21_25 3.33_26 3.33_27 3.37_28 3.44_29 3.54_30 3.54_31
##       2       2       2       2       2       2       2       2       2       2
## 3.56_32 3.61_33 3.71_34 3.79_35 3.84_36 3.39_37  3.4_38 3.41_39 3.43_40 3.44_41
##       2       2       2       1       1       1       1       1       1       1
## 3.46_42 3.47_43 3.49_44  3.5_45  3.5_46 3.52_47 3.53_48 3.53_49 3.54_50 3.55_51
##       1       1       1       1       1       1       1       1       1       1
## 3.55_52 3.56_53 3.56_54 3.57_55 3.58_56  3.5_58 3.65_59 3.66_60 3.69_61  3.7_62
##       1       1       1       1       1       3       3       3       3       3
##  3.7_63 3.78_64  3.8_65  3.8_66 3.87_67 3.88_68  3.9_69 3.92_70 3.93_71  3.8_72
##       3       3       3       3       3       3       3       3       3       3
## 2.77_73  2.9_74 2.91_75    3_76  3.1_77 3.11_78 3.22_79 3.32_80 3.56_81 3.74_82
##       2       2       2       2       2       2       2       2       1       2
## 3.29_83 3.42_84 3.45_85 3.51_86 3.52_87
##       1       1       1       1       1
##
## Within cluster sum of squares by cluster:
## [1] 2317.7493 3902.5662  276.4358
##  (between_SS / total_SS =  98.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

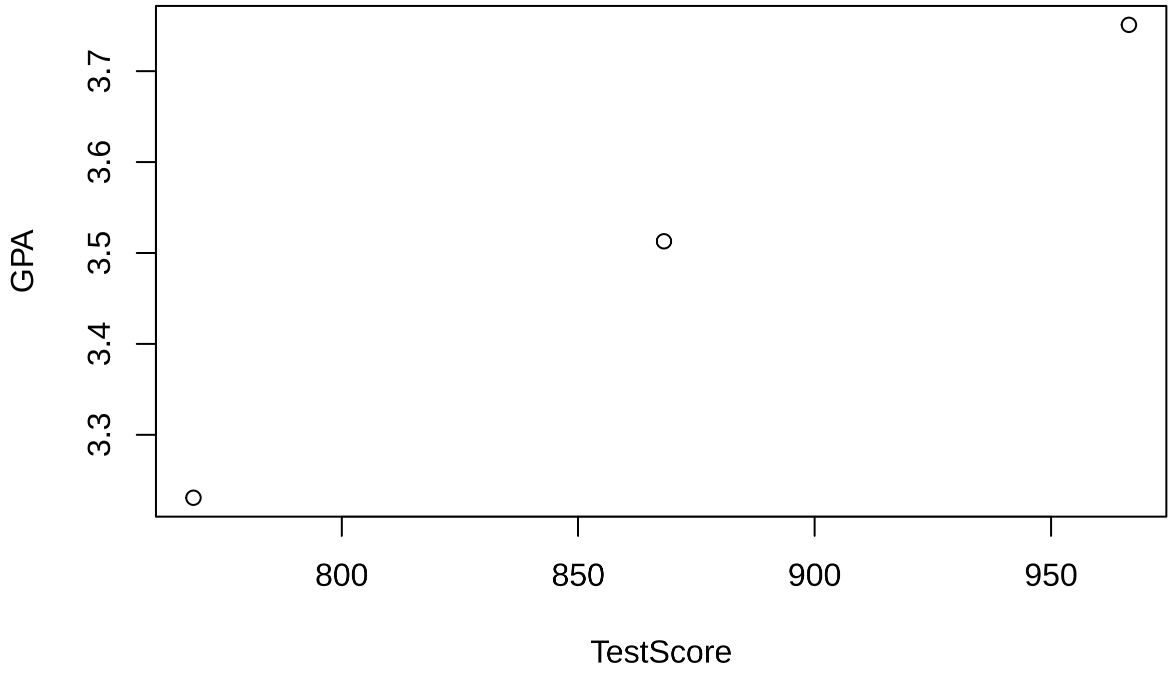
```r
## This shows the cluster centroid means
kmeansFIT$centers
```

```
##   TestScore      GPA
## 1  868.1429 3.512857
## 2  768.6400 3.230800
## 3  966.4688 3.750938
```

```r
plot(kmeansFIT$cluster)
```



```r
plot(kmeansFIT$centers)
```

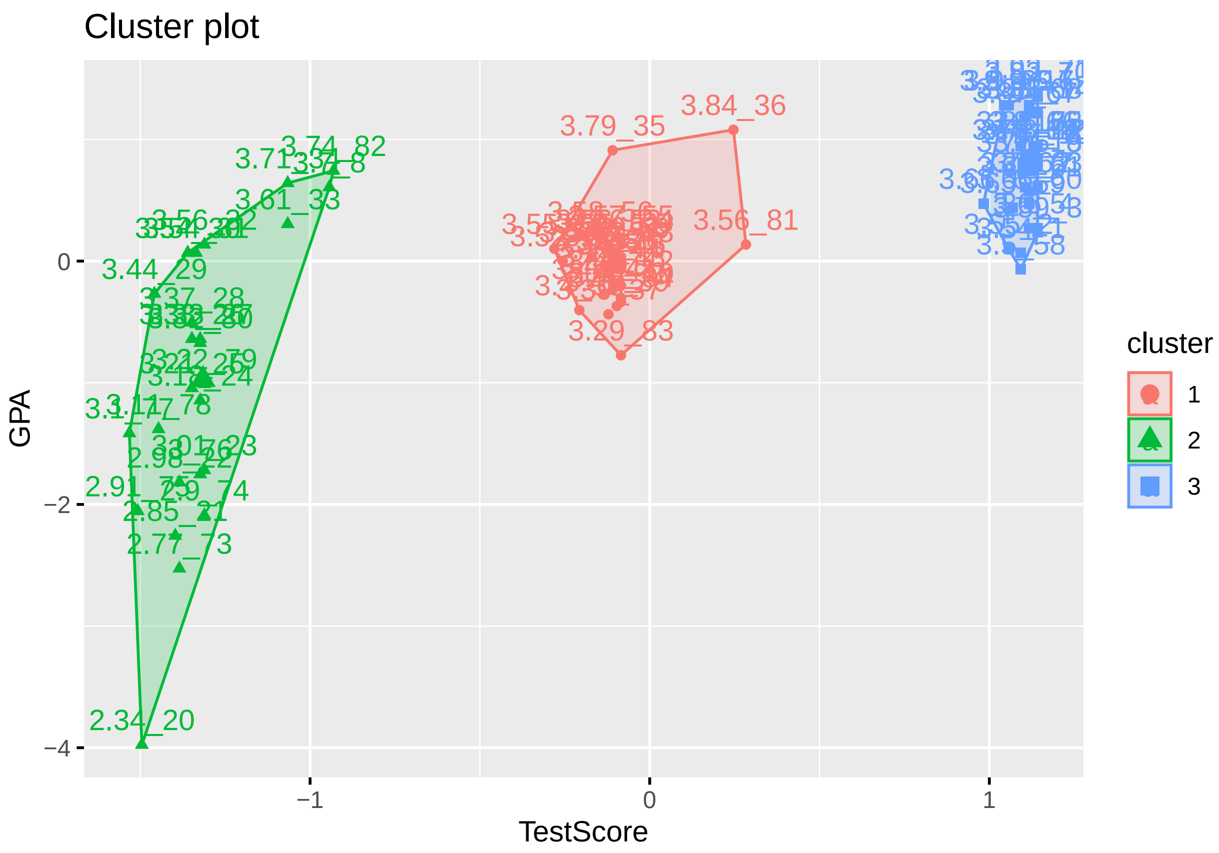


```
############################################
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.5.3
```

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

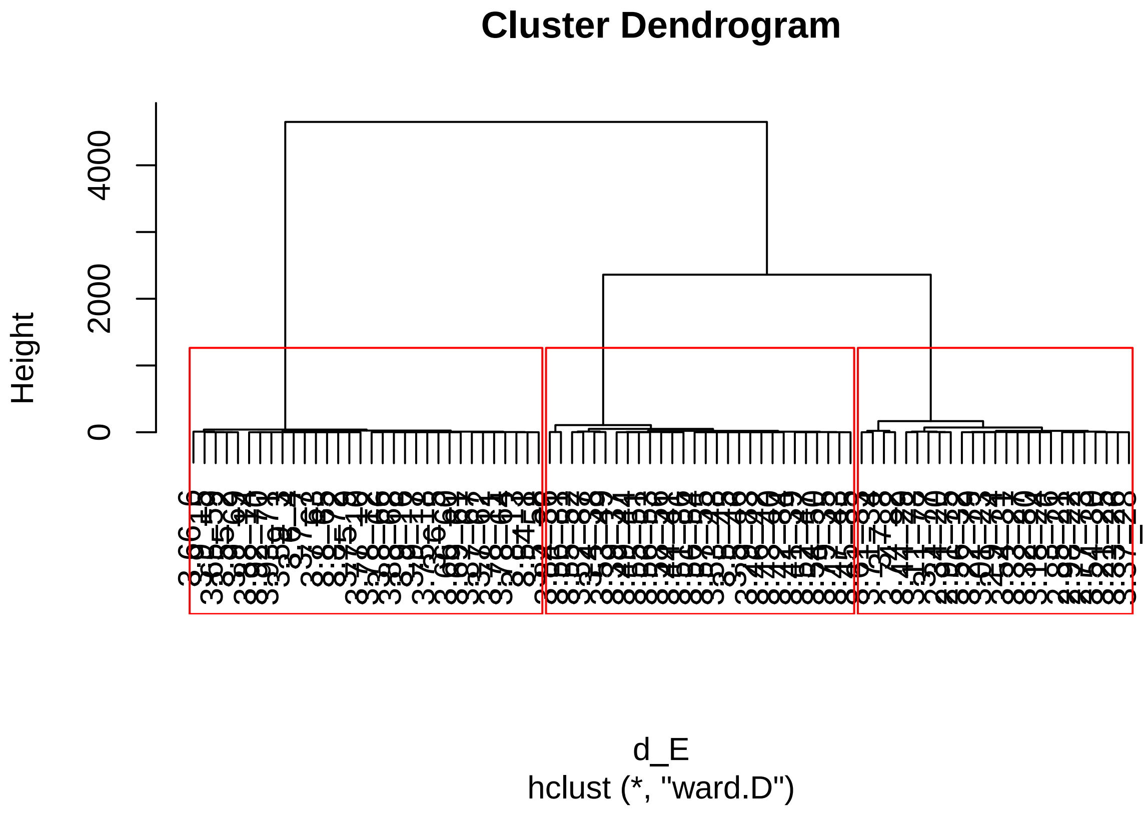
```
fviz_cluster(list
                  (data = C_Data,
                  cluster = kmeansFIT$cluster,
                  show.clust.cent = TRUE,
                  labelsize = 1,
                  main = "Cluster of Admissions Data",
                  ellipse.type = "norm",
                  outlier.shape = 10,
                  outlier.pointsize = 5,
                  outlier.labelsize = 5,
                    pointsize = .5,
                    repel = TRUE
                    ))
```



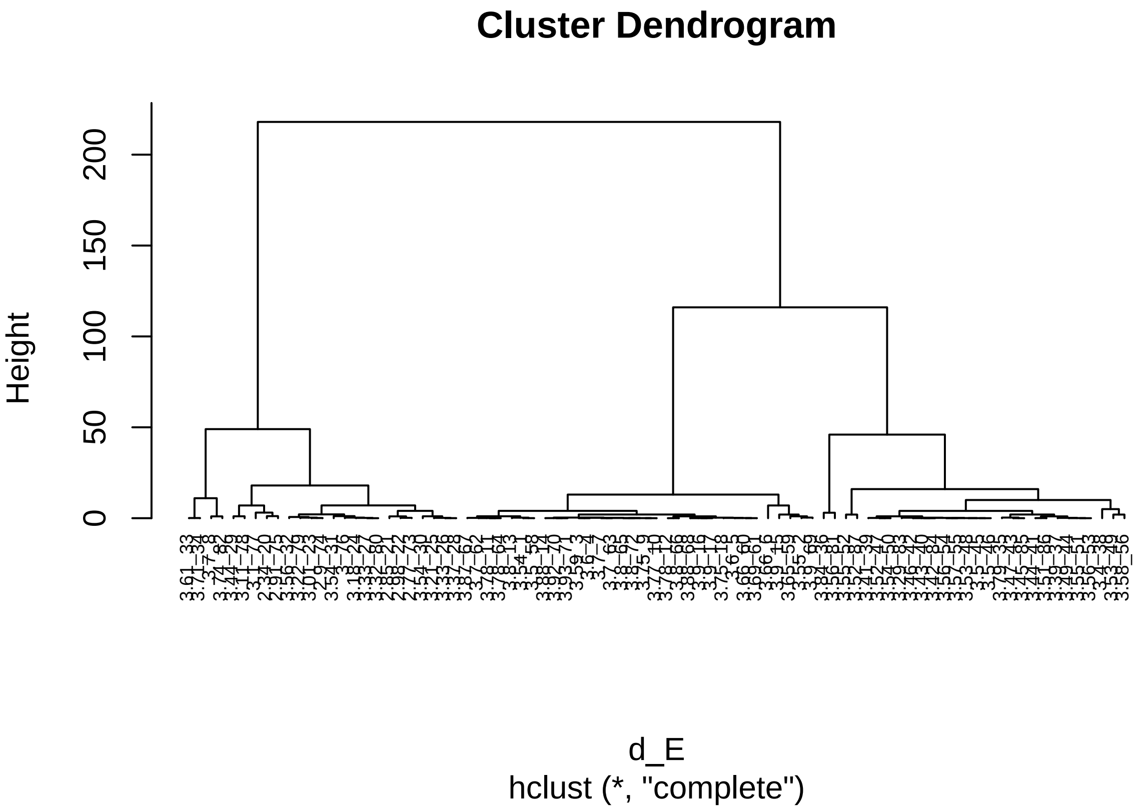
```
####################
## Hierarchical clustering
################################################
d_E <- dist(C_Data, method = "euclidean") # distance matrix

fit1 <- hclust(d_E, method="ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

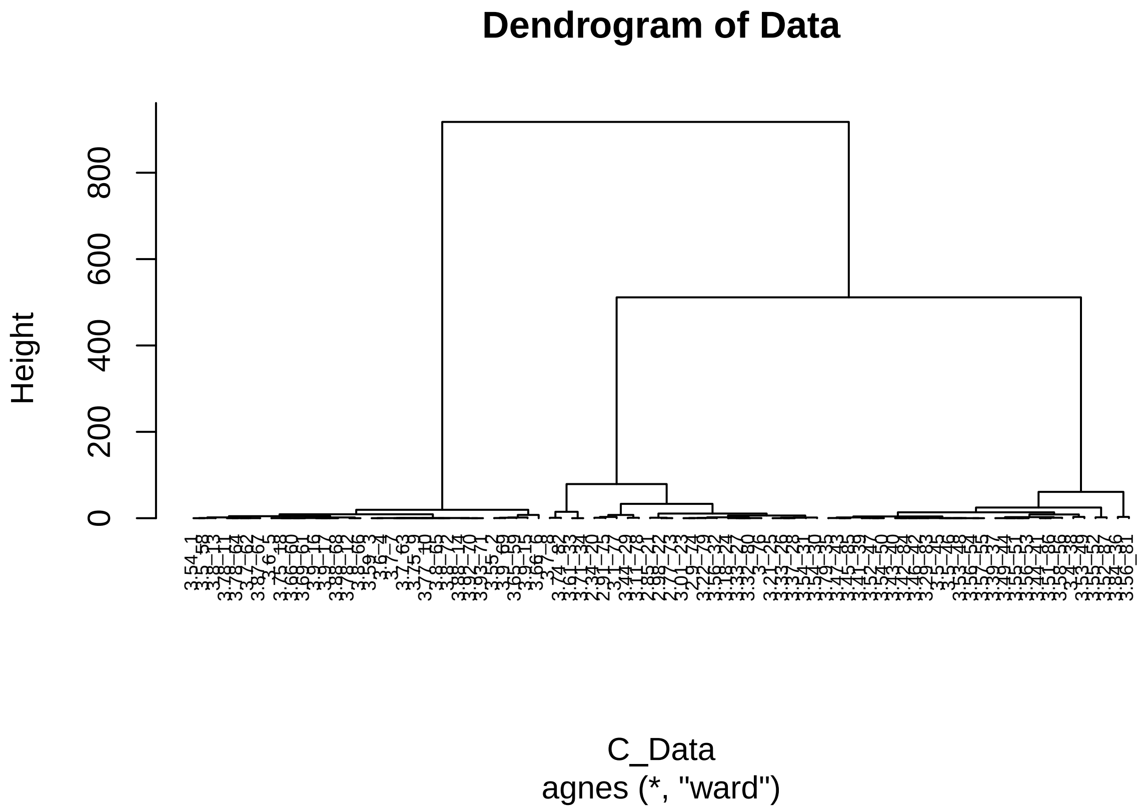
```
plot(fit1) # display dendogram
groups <- cutree(fit1, k=3) # cut tree into 3 clusters
# draw dendogram with red borders around the 3 clusters
rect.hclust(fit1, k=3, border="red")
```



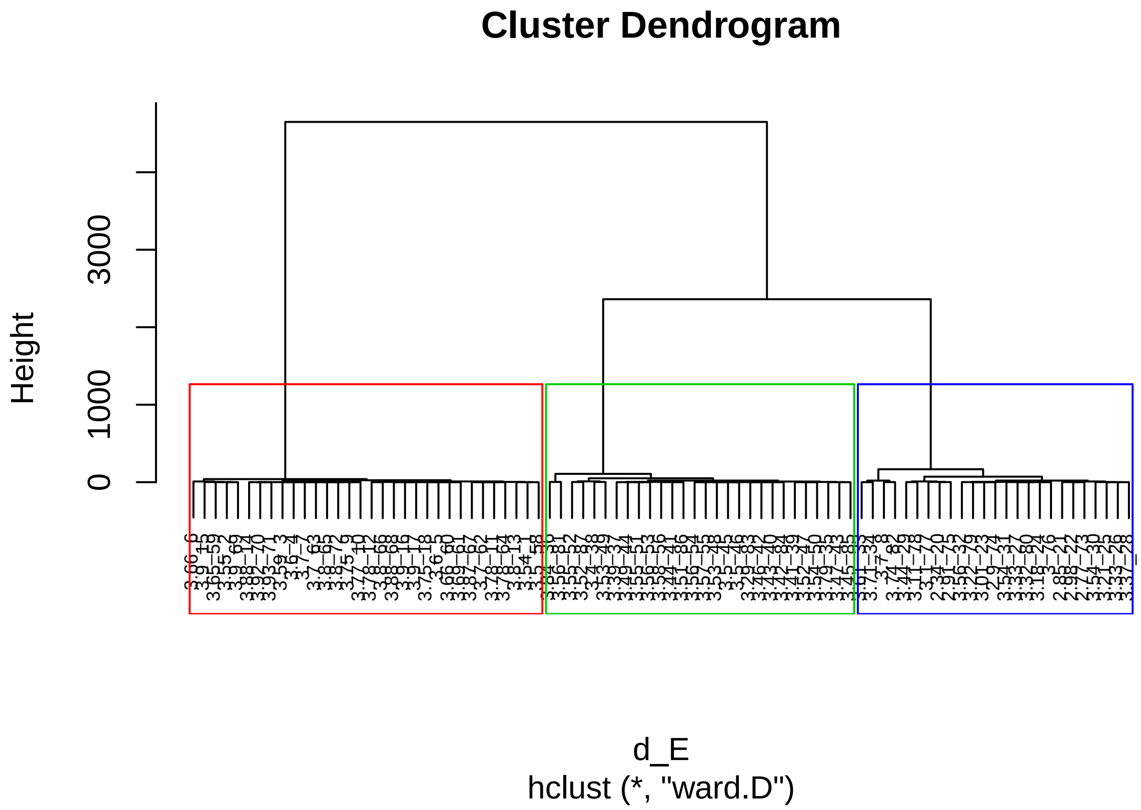
```r
# Hierarchical clustering using Complete Linkage
fit2 <- hclust(d_E, method = "complete" )

# Plot the obtained dendrogram
plot(fit2, cex = 0.6, hang = -1)
```



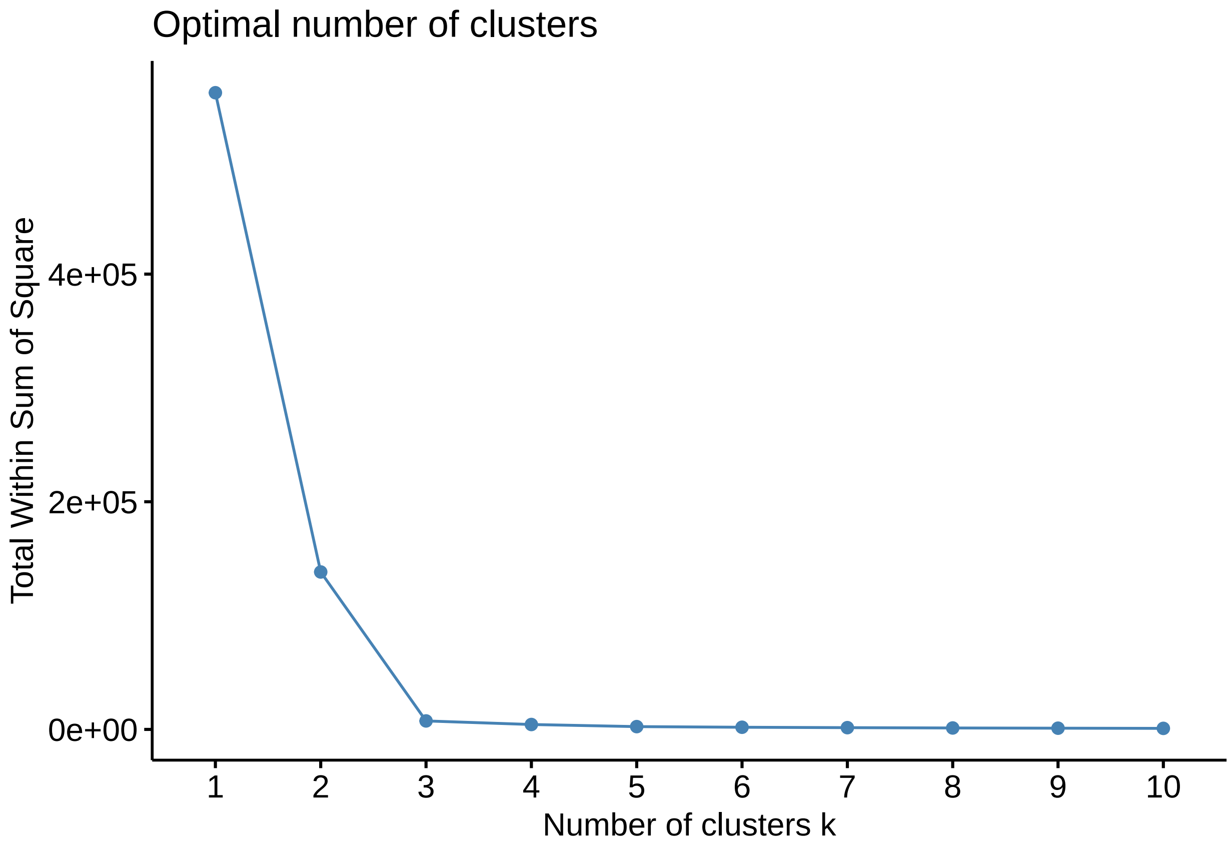
```
fit3 <- agnes(C_Data, method = "ward")
pltree(fit3, cex = 0.6,
       hang = -1, main = "Dendrogram of Data")
```



```
plot(fit1, cex = 0.6)
rect.hclust(fit1, k = 3, border = 2:5)
```



```
####################
## Determine optimal number of clusters....
##---------------------------------------------
fviz_nbclust(MyClusterData, FUN = hcut, method = "wss")
```



```
## We can see that we are right to use k = 3


##################################################
##
##          Association Rule Mining
##
##################################################
##
## We first need to format the data - BUT -
## we need to know what we are trying to associate??

head(MyData)
```

```
##   Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1    Admit Female 2020-01-11    Florida 3.54     0.7       965           94
## 2    Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3    Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4    Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5    Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6    Admit Female 2019-11-03 California 3.66     0.9       956           89
##   VolunteerLevel
## 1              1
## 2              0
## 3              0
## 4              2
## 5              2
## 6              1
```

```
## With ARM - the idea is to see if certain words
## seems more associated with a greater probability.
## We CANNOT do this with numeric data and so we must
## discretize or remove all numeric or date data.
## We must end up with ONLY categorical data.

names(MyData)
```

```
## [1] "Decision"       "Gender"         "DateSub"        "State"
## [5] "GPA"            "WorkExp"        "TestScore"      "WritingScore"
## [9] "VolunteerLevel"
```

```
## Columns that are fine as-is are Decision, Gender, and
## VolunteerLevel. All other columns must be discretized.

sort(MyData$DateSub)
```

```
##  [1] "2019-09-13" "2019-10-03" "2019-10-07" "2019-10-10" "2019-10-14"
##  [6] "2019-10-19" "2019-10-25" "2019-10-30" "2019-10-31" "2019-11-01"
## [11] "2019-11-02" "2019-11-03" "2019-11-04" "2019-11-04" "2019-11-07"
## [16] "2019-11-07" "2019-11-08" "2019-11-09" "2019-11-10" "2019-11-15"
## [21] "2019-11-16" "2019-11-17" "2019-11-19" "2019-11-21" "2019-11-25"
## [26] "2019-11-26" "2019-11-27" "2019-11-28" "2019-11-30" "2019-12-01"
## [31] "2019-12-03" "2019-12-03" "2019-12-04" "2019-12-04" "2019-12-06"
## [36] "2019-12-08" "2019-12-09" "2019-12-10" "2019-12-11" "2019-12-20"
## [41] "2019-12-20" "2019-12-21" "2019-12-23" "2019-12-24" "2019-12-25"
## [46] "2019-12-27" "2019-12-28" "2019-12-29" "2019-12-30" "2019-12-30"
## [51] "2019-12-31" "2020-01-05" "2020-01-05" "2020-01-10" "2020-01-10"
## [56] "2020-01-11" "2020-01-11" "2020-01-12" "2020-01-12" "2020-01-12"
## [61] "2020-01-14" "2020-01-15" "2020-01-15" "2020-01-17" "2020-01-22"
## [66] "2020-01-23" "2020-01-25" "2020-01-25" "2020-01-28" "2020-01-29"
## [71] "2020-01-30" "2020-01-31" "2020-02-01" "2020-02-01" "2020-02-04"
## [76] "2020-02-07" "2020-02-10" "2020-02-11" "2020-02-15" "2020-02-15"
## [81] "2020-02-16" "2020-02-19" "2020-02-27" "2020-03-06" "2020-03-20"
```

```
## New DF
MyARM_Data<-data.frame()
str(MyARM_Data)
```

```
## 'data.frame':    0 obs. of  0 variables
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.5.3

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## First, convert all dates to a month
(Temp4<-MyData %>% mutate(month = month(DateSub)))
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1     Admit Female 2020-01-11    Florida 3.54     0.7       965           94
## 2     Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3     Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4     Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5     Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6     Admit Female 2019-11-03 California 3.66     0.9       956           89
## 7     Admit Female 2019-11-08 California 3.70     1.2       969           94
## 8     Admit Female 2019-10-07 California 3.70     2.7       799           97
## 9     Admit Female 2019-10-10   Colorado 3.75     1.1       969           93
## 10    Admit Female 2020-01-15    Florida 3.77     1.4       969           99
## 11    Admit Female 2019-10-31 California 3.78     8.7       966           91
## 12    Admit Female 2019-10-30       Utah 3.78     1.2       968           87
## 13    Admit Female 2019-10-14    Florida 3.80     1.9       965           94
## 14    Admit Female 2019-11-04   Colorado 3.88     1.0       969           93
## 15    Admit Female 2019-12-20    Florida 3.90     4.7       961           93
## 16    Admit Female 2019-10-25   Colorado 3.90     3.8       967           98
## 17    Admit Female 2019-12-28    Florida 3.90     0.0       967           88
## 18    Admit Female 2020-01-10 California 3.75     2.8       967           95
## 19  Decline Female 2020-01-14 California 2.34     0.8       754           94
## 20  Decline Female 2020-01-31    Colorado 2.85     4.6       762           94
## 21  Decline Female 2020-01-10    Colorado 2.98     0.7       763           94
## 22  Decline Female 2020-01-25        Utah 3.01     1.4       769           94
## 23  Decline Female 2020-02-27     Florida 3.18     1.4       768           94
## 24  Decline Female 2019-12-31    Virginia 3.21     1.7       766           94
## 25  Decline Female 2020-03-06     Florida 3.33     1.6       766           94
## 26  Decline Female 2020-02-07        Utah 3.33     0.8       768           94
## 27  Decline Female 2019-12-03    Virginia 3.37     0.9       766           94
## 28  Decline Female 2019-11-30    Colorado 3.44     3.2       757           94
## 29  Decline Female 2020-01-12     Florida 3.54     0.9       765           94
## 30  Decline Female 2020-02-15     Florida 3.54     1.1       767           94
## 31  Decline Female 2019-12-10     Florida 3.56     1.7       769           94
## 32  Decline Female 2020-01-22     Florida 3.61     1.3       789           94
## 33  Decline Female 2019-12-04     Florida 3.71     0.7       789           94
## 34  Decline Female 2019-11-25     Florida 3.79     1.4       867           94
## 35  Decline Female 2020-01-12     Florida 3.84     2.7       896           89
## 36 Waitlist Female 2019-12-30        Utah 3.39     1.8       866           94
## 37 Waitlist Female 2020-02-19     Florida 3.40     1.9       859           88
## 38 Waitlist Female 2019-12-09     Alabama 3.41     1.2       868           85
## 39 Waitlist Female 2019-12-23    Colorado 3.43     1.5       869           85
## 40 Waitlist Female 2020-01-29     Florida 3.44     7.2       865           94
## 41 Waitlist Female 2020-02-16     Florida 3.46     1.9       869           89
## 42 Waitlist Female 2020-01-17  California 3.47     2.2       867           94
## 43 Waitlist Female 2019-12-27    Colorado 3.49     1.3       866           86
## 44 Waitlist Female 2020-02-04     Florida 3.50     1.7       869           94
## 45 Waitlist Female 2019-12-04    Colorado 3.50     3.5       869           94
## 46 Waitlist Female 2020-01-23     Florida 3.52     0.7       868           94
## 47 Waitlist Female 2020-01-30     Florida 3.53     1.7       869           87
## 48 Waitlist Female 2019-11-28     Vermont 3.53     3.3       862           85
## 49 Waitlist Female 2019-11-04    Colorado 3.54     1.2       868           94
## 50 Waitlist Female 2019-12-08    New York 3.55     2.2       866           94
## 51 Waitlist Female 2019-12-11     Florida 3.55     2.0       853           94
## 52 Waitlist Female 2019-12-06     Georgia 3.56     1.0       866           89
## 53 Waitlist Female 2019-11-17     Florida 3.56     1.3       869           94
## 54 Waitlist Female 2019-11-15        Utah 3.57     1.4       869           94
## 55 Waitlist Female 2019-11-09        Utah 3.58     0.9       864           94
## 56    Admit   Male 2020-01-25     Florida 3.50     0.7       965           91
## 57    Admit   Male 2019-11-10    Colorado 3.65     1.7       963           90
## 58    Admit   Male 2019-12-21     Florida 3.66     2.2       967           91
## 59    Admit   Male 2019-12-03  California 3.69     3.2       967           93
## 60    Admit   Male 2019-11-26  California 3.70     1.4       966           94
## 61    Admit   Male 2019-11-01     Florida 3.70     3.7       969           99
## 62    Admit   Male 2019-11-16    Colorado 3.78     1.2       966           94
## 63    Admit   Male 2019-12-20     Florida 3.80     1.4       969           97
## 64    Admit   Male 2019-11-27     Florida 3.80     1.7       968           91
## 65    Admit   Male 2019-11-19  California 3.87     1.7       966           97
## 66    Admit   Male 2019-10-19  California 3.88     1.5       967           95
## 67    Admit   Male 2019-09-13  California 3.90     6.7       962          100
## 68    Admit   Male 2019-10-03    Colorado 3.92     1.2       969           95
## 69    Admit   Male 2019-11-02     Florida 3.93     0.8       969           99
## 70    Admit   Male 2019-12-24    Colorado 3.80     0.8       969           93
## 71  Decline   Male 2020-02-15    Virginia 2.77     3.7       763           94
## 72  Decline   Male 2020-02-01      Oregon 2.90     0.9       769           87
## 73  Decline   Male 2020-02-11 mississippi 2.91     6.2       753           94
## 74  Decline   Male 2020-01-15    Colorado 3.00     1.2       768           94
## 75  Decline   Male 2020-03-20     Florida 3.10     1.9       751           94
## 76  Decline   Male 2020-02-01    Colorado 3.11     1.7       758           94
## 77  Decline   Male 2020-01-05    Virginia 3.22     3.2       769           94
## 78  Decline   Male 2019-12-25    Colorado 3.32     1.7       768           94
## 79  Decline   Male 2020-01-05     Florida 3.56     3.7       899           94
## 80  Decline   Male 2019-12-30     Florida 3.74     1.3       800           94
## 81 Waitlist   Male 2020-01-28     Florida 3.29     1.2       869           94
## 82 Waitlist   Male 2019-12-01  California 3.42     0.7       869           94
## 83 Waitlist   Male 2020-02-10     Florida 3.45     4.7       867           94
## 84 Waitlist   Male 2019-12-29     Florida 3.51     3.4       865           88
## 85 Waitlist   Male 2019-11-07  California 3.52     2.7       855           87
##    VolunteerLevel month
## 1               1     1
## 2               0     1
## 3               0     1
## 4               2    11
## 5               2    11
## 6               1    11
## 7               2    11
## 8               5    10
## 9               0    10
## 10              4     1
## 11              2    10
## 12              2    10
## 13              5    10
## 14              4    11
## 15              1    12
## 16              3    10
## 17              0    12
## 18              3     1
## 19              1     1
## 20              4     1
## 21              1     1
## 22              2     1
## 23              0     2
## 24              5    12
## 25              5     3
## 26              1     2
## 27              2    12
## 28              3    11
## 29              0     1
## 30              4     2
## 31              4    12
## 32              5     1
## 33              4    12
## 34              2    11
## 35              1     1
## 36              5    12
## 37              4     2
## 38              0    12
## 39              1    12
## 40              2     1
## 41              0     2
## 42               5     1
## 43               5    12
## 44               2     2
## 45               4    12
## 46               4     1
## 47               2     1
## 48               1    11
## 49               2    11
## 50               1    12
## 51               1    12
## 52               1    12
## 53               3    11
## 54               0    11
## 55               5    11
## 56               1     1
## 57               1    11
## 58               2    12
## 59               3    12
## 60               0    11
## 61               2    11
## 62               4    11
## 63               4    12
## 64               3    11
## 65               5    11
## 66               5    10
## 67               0     9
## 68               3    10
## 69               4    11
## 70               1    12
## 71               5     2
## 72               4     2
## 73               1     2
## 74               1     1
## 75               0     3
## 76               3     2
## 77               1     1
## 78               4    12
## 79               3     1
## 80               4    12
## 81               4     1
## 82               3    12
## 83               3     2
## 84               3    12
## 85               3    11
```

```
## Then, cut by month
(Temp5 <-
  cut(Temp4$month,
      breaks=c(0, 1, 3, 11, 12),
      labels=c("Jan","Feb_Mar", "Nov","Dec")))
```

```
##  [1] Jan     Jan     Jan     Nov     Nov     Nov     Nov     Nov     Nov
## [10] Jan     Nov     Nov     Nov     Nov     Dec     Nov     Dec     Jan
## [19] Jan     Jan     Jan     Jan     Feb_Mar Dec     Feb_Mar Feb_Mar Dec
## [28] Nov     Jan     Feb_Mar Dec     Jan     Dec     Nov     Jan     Dec    
## [37] Feb_Mar Dec     Dec     Jan     Feb_Mar Jan     Dec     Feb_Mar Dec    
## [46] Jan     Jan     Nov     Nov     Dec     Dec     Dec     Nov     Nov    
## [55] Nov     Jan     Nov     Dec     Dec     Nov     Nov     Nov     Dec    
## [64] Nov     Nov     Nov     Nov     Nov     Nov     Dec     Feb_Mar Feb_Mar
## [73] Feb_Mar Jan     Feb_Mar Feb_Mar Jan     Dec     Jan     Dec     Jan    
## [82] Dec     Feb_Mar Dec     Nov    
## Levels: Jan Feb_Mar Nov Dec
```

```
## Next, convert the GPA

(Temp6 <-
    cut(MyData$GPA,
        breaks=c(0, 3.1, 3.4, 3.59, 3.8, 4),
        labels=c("C_B-", "B","A-", "A", "A+")))
```

```
##  [1] A-   A-   A-   A    A    A    A    A    A    A    A    A    A    A+   A+  
## [16] A+   A+   A    C_B- C_B- C_B- C_B- B    B    B    B    B    A-   A-   A-  
## [31] A-   A    A    A    A+   B    B    A-   A-   A-   A-   A-   A-   A-   A-  
## [46] A-   A-   A-   A-   A-   A-   A-   A-   A-   A-   A-   A    A    A    A   
## [61] A    A    A    A    A+   A+   A+   A+   A+   A    C_B- C_B- C_B- C_B- C_B-
## [76] B    B    B    A-   A    B    A-   A-   A-   A-  
## Levels: C_B- B A- A A+
```

```
(MyARM_Data<-cbind(as.data.frame(Temp6), as.data.frame(Temp5)))
```

```
##    Temp6   Temp5
## 1     A-     Jan
## 2     A-     Jan
## 3     A-     Jan
## 4      A     Nov
## 5      A     Nov
## 6      A     Nov
## 7      A     Nov
## 8      A     Nov
## 9      A     Nov
## 10     A     Jan
## 11     A     Nov
## 12     A     Nov
## 13     A     Nov
## 14    A+     Nov
## 15    A+     Dec
## 16    A+     Nov
## 17    A+     Dec
## 18     A     Jan
## 19  C_B-     Jan
## 20  C_B-     Jan
## 21  C_B-     Jan
## 22  C_B-     Jan
## 23     B Feb_Mar
## 24     B     Dec
## 25     B Feb_Mar
## 26     B Feb_Mar
## 27     B     Dec
## 28    A-     Nov
## 29    A-     Jan
## 30    A- Feb_Mar
## 31    A-     Dec
## 32     A     Jan
## 33     A     Dec
## 34     A     Nov
## 35    A+     Jan
## 36     B     Dec
## 37     B Feb_Mar
## 38    A-     Dec
## 39    A-     Dec
## 40    A-     Jan
## 41    A- Feb_Mar
## 42    A-     Jan
## 43    A-     Dec
## 44    A- Feb_Mar
## 45    A-     Dec
## 46    A-     Jan
## 47    A-     Jan
## 48    A-     Nov
## 49    A-     Nov
## 50    A-     Dec
## 51    A-     Dec
## 52    A-     Dec
## 53    A-     Nov
## 54    A-     Nov
## 55    A-     Nov
## 56    A-     Jan
## 57     A     Nov
## 58     A     Dec
## 59     A     Dec
## 60     A     Nov
## 61     A     Nov
## 62     A     Nov
## 63     A     Dec
## 64     A     Nov
## 65    A+     Nov
## 66    A+     Nov
## 67    A+     Nov
## 68    A+     Nov
## 69    A+     Nov
## 70     A     Dec
## 71  C_B- Feb_Mar
## 72  C_B- Feb_Mar
## 73  C_B- Feb_Mar
## 74  C_B-     Jan
## 75  C_B- Feb_Mar
## 76     B Feb_Mar
## 77     B     Jan
## 78     B     Dec
## 79    A-     Jan
## 80     A     Dec
## 81     B     Jan
## 82    A-     Dec
## 83    A- Feb_Mar
## 84    A-     Dec
## 85    A-     Nov
```

```
names(MyARM_Data)<-c("GPALevel","SubDate")
MyARM_Data
```

```
##    GPALevel SubDate
## 1        A-     Jan
## 2        A-     Jan
## 3        A-     Jan
## 4         A     Nov
## 5         A     Nov
## 6         A     Nov
## 7         A     Nov
## 8         A     Nov
## 9         A     Nov
## 10        A     Jan
## 11        A     Nov
## 12        A     Nov
## 13        A     Nov
## 14       A+     Nov
## 15       A+     Dec
## 16       A+     Nov
## 17       A+     Dec
## 18        A     Jan
## 19     C_B-     Jan
## 20     C_B-     Jan
## 21     C_B-     Jan
## 22     C_B-     Jan
## 23        B Feb_Mar
## 24        B     Dec
## 25        B Feb_Mar
## 26        B Feb_Mar
## 27        B     Dec
## 28       A-     Nov
## 29       A-     Jan
## 30       A- Feb_Mar
## 31       A-     Dec
## 32        A     Jan
## 33        A     Dec
## 34        A     Nov
## 35       A+     Jan
## 36        B     Dec
## 37        B Feb_Mar
## 38       A-     Dec
## 39       A-     Dec
## 40       A-     Jan
## 41       A- Feb_Mar
## 42       A-     Jan
## 43       A-     Dec
## 44       A- Feb_Mar
## 45        A-     Dec
## 46        A-     Jan
## 47        A-     Jan
## 48        A-     Nov
## 49        A-     Nov
## 50        A-     Dec
## 51        A-     Dec
## 52        A-     Dec
## 53        A-     Nov
## 54        A-     Nov
## 55        A-     Nov
## 56        A-     Jan
## 57         A     Nov
## 58         A     Dec
## 59         A     Dec
## 60         A     Nov
## 61         A     Nov
## 62         A     Nov
## 63         A     Dec
## 64         A     Nov
## 65        A+     Nov
## 66        A+     Nov
## 67        A+     Nov
## 68        A+     Nov
## 69        A+     Nov
## 70         A     Dec
## 71      C_B- Feb_Mar
## 72      C_B- Feb_Mar
## 73      C_B- Feb_Mar
## 74      C_B-     Jan
## 75      C_B- Feb_Mar
## 76         B Feb_Mar
## 77         B     Jan
## 78         B     Dec
## 79        A-     Jan
## 80         A     Dec
## 81         B     Jan
## 82        A-     Dec
## 83        A- Feb_Mar
## 84        A-     Dec
## 85        A-     Nov
```

```
## Next, discretize the others...
(Temp7 <-
    cut(MyData$TestScore,
        breaks=c(0, 700, 800, 900, 1000),
        labels=c("Low", "Medium","High", "VeryHigh")))
```

```
##  [1] VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh Medium
##  [9] VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh
## [17] VeryHigh VeryHigh Medium   Medium   Medium   Medium   Medium   Medium
## [25] Medium   Medium   Medium   Medium   Medium   Medium   Medium   Medium
## [33] Medium   High     High     High     High     High     High     High
## [41] High     High     High     High     High     High     High     High
## [49] High     High     High     High     High     High     High     VeryHigh
## [57] VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh
## [65] VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh VeryHigh Medium   Medium
## [73] Medium   Medium   Medium   Medium   Medium   Medium   High     Medium
## [81] High     High     High     High     High
## Levels: Low Medium High VeryHigh
```

```
(MyARM_Data<-cbind(as.data.frame(Temp7), MyARM_Data))
```

```
##       Temp7 GPALevel SubDate
## 1  VeryHigh       A-     Jan
## 2  VeryHigh       A-     Jan
## 3  VeryHigh       A-     Jan
## 4  VeryHigh        A     Nov
## 5  VeryHigh        A     Nov
## 6  VeryHigh        A     Nov
## 7  VeryHigh        A     Nov
## 8    Medium        A     Nov
## 9  VeryHigh        A     Nov
## 10 VeryHigh        A     Jan
## 11 VeryHigh        A     Nov
## 12 VeryHigh        A     Nov
## 13 VeryHigh        A     Nov
## 14 VeryHigh       A+     Nov
## 15 VeryHigh       A+     Dec
## 16 VeryHigh       A+     Nov
## 17 VeryHigh       A+     Dec
## 18 VeryHigh        A     Jan
## 19   Medium     C_B-     Jan
## 20   Medium     C_B-     Jan
## 21   Medium     C_B-     Jan
## 22   Medium     C_B-     Jan
## 23   Medium        B Feb_Mar
## 24   Medium        B     Dec
## 25   Medium        B Feb_Mar
## 26   Medium        B Feb_Mar
## 27   Medium        B     Dec
## 28   Medium       A-     Nov
## 29   Medium       A-     Jan
## 30   Medium       A- Feb_Mar
## 31   Medium       A-     Dec
## 32   Medium        A     Jan
## 33   Medium        A     Dec
## 34     High        A     Nov
## 35     High       A+     Jan
## 36     High        B     Dec
## 37     High        B Feb_Mar
## 38     High       A-     Dec
## 39     High       A-     Dec
## 40     High       A-     Jan
## 41     High       A- Feb_Mar
## 42     High       A-     Jan
## 43     High       A-     Dec
## 44     High       A- Feb_Mar
## 45     High       A-     Dec
## 46     High       A-     Jan
## 47     High       A-     Jan
## 48     High       A-     Nov
## 49     High       A-     Nov
## 50     High       A-     Dec
## 51     High       A-     Dec
## 52     High       A-     Dec
## 53     High       A-     Nov
## 54     High       A-     Nov
## 55     High       A-     Nov
## 56 VeryHigh       A-     Jan
## 57 VeryHigh        A     Nov
## 58 VeryHigh        A     Dec
## 59 VeryHigh        A     Dec
## 60 VeryHigh        A     Nov
## 61 VeryHigh        A     Nov
## 62 VeryHigh        A     Nov
## 63 VeryHigh        A     Dec
## 64 VeryHigh        A     Nov
## 65 VeryHigh       A+     Nov
## 66 VeryHigh       A+     Nov
## 67 VeryHigh       A+     Nov
## 68 VeryHigh       A+     Nov
## 69 VeryHigh       A+     Nov
## 70 VeryHigh        A     Dec
## 71   Medium     C_B- Feb_Mar
## 72   Medium     C_B- Feb_Mar
## 73   Medium     C_B- Feb_Mar
## 74   Medium     C_B-     Jan
## 75   Medium     C_B- Feb_Mar
## 76   Medium        B Feb_Mar
## 77   Medium        B     Jan
## 78   Medium        B     Dec
## 79     High       A-     Jan
## 80   Medium        A     Dec
## 81     High        B     Jan
## 82     High       A-     Dec
## 83     High       A- Feb_Mar
## 84     High       A-     Dec
## 85     High       A-     Nov
```

```
names(MyARM_Data)<-c("TestScore", "GPALevel","SubDate")
MyARM_Data
```

```
##    TestScore GPALevel SubDate
## 1   VeryHigh       A-     Jan
## 2   VeryHigh       A-     Jan
## 3   VeryHigh       A-     Jan
## 4   VeryHigh        A     Nov
## 5   VeryHigh        A     Nov
## 6   VeryHigh        A     Nov
## 7   VeryHigh        A     Nov
## 8     Medium        A     Nov
## 9   VeryHigh        A      Nov
## 10  VeryHigh        A      Jan
## 11  VeryHigh        A      Nov
## 12  VeryHigh        A      Nov
## 13  VeryHigh        A      Nov
## 14  VeryHigh       A+      Nov
## 15  VeryHigh       A+      Dec
## 16  VeryHigh       A+      Nov
## 17  VeryHigh       A+      Dec
## 18  VeryHigh        A      Jan
## 19    Medium     C_B-      Jan
## 20    Medium     C_B-      Jan
## 21    Medium     C_B-      Jan
## 22    Medium     C_B-      Jan
## 23    Medium        B  Feb_Mar
## 24    Medium        B      Dec
## 25    Medium        B  Feb_Mar
## 26    Medium        B  Feb_Mar
## 27    Medium        B      Dec
## 28    Medium       A-      Nov
## 29    Medium       A-      Jan
## 30    Medium       A-  Feb_Mar
## 31    Medium       A-      Dec
## 32    Medium        A      Jan
## 33    Medium        A      Dec
## 34      High        A      Nov
## 35      High       A+      Jan
## 36      High        B      Dec
## 37      High        B  Feb_Mar
## 38      High       A-      Dec
## 39      High       A-      Dec
## 40      High       A-      Jan
## 41      High       A-  Feb_Mar
## 42      High       A-      Jan
## 43      High       A-      Dec
## 44      High       A-  Feb_Mar
## 45      High       A-      Dec
## 46      High       A-      Jan
## 47      High       A-      Jan
## 48      High       A-      Nov
## 49      High       A-      Nov
## 50      High       A-      Dec
## 51      High       A-      Dec
## 52      High       A-      Dec
## 53      High       A-      Nov
## 54      High       A-      Nov
## 55      High       A-      Nov
## 56  VeryHigh       A-      Jan
## 57  VeryHigh        A      Nov
## 58  VeryHigh        A      Dec
## 59  VeryHigh        A      Dec
## 60  VeryHigh        A      Nov
## 61  VeryHigh        A      Nov
## 62  VeryHigh        A      Nov
## 63  VeryHigh        A     Dec
## 64  VeryHigh        A     Nov
## 65  VeryHigh       A+     Nov
## 66  VeryHigh       A+     Nov
## 67  VeryHigh       A+     Nov
## 68  VeryHigh       A+     Nov
## 69  VeryHigh       A+     Nov
## 70  VeryHigh        A     Dec
## 71    Medium     C_B- Feb_Mar
## 72    Medium     C_B- Feb_Mar
## 73    Medium     C_B- Feb_Mar
## 74    Medium     C_B-     Jan
## 75    Medium     C_B- Feb_Mar
## 76    Medium        B Feb_Mar
## 77    Medium        B     Jan
## 78    Medium        B     Dec
## 79      High       A-     Jan
## 80    Medium        A     Dec
## 81      High        B     Jan
## 82      High       A-     Dec
## 83      High       A- Feb_Mar
## 84      High       A-     Dec
## 85      High       A-     Nov
```

```r
## Let's include other categorical variables as well

names(MyData)
```

```
## [1] "Decision"       "Gender"         "DateSub"        "State"
## [5] "GPA"            "WorkExp"        "TestScore"      "WritingScore"
## [9] "VolunteerLevel"
```

```r
(MyARM_Data<-cbind(MyARM_Data, MyData$Decision, MyData$Gender, MyData$VolunteerLevel))
```

```
##    TestScore GPALevel SubDate MyData$Decision MyData$Gender
## 1   VeryHigh       A-     Jan           Admit        Female
## 2   VeryHigh       A-     Jan           Admit        Female
## 3   VeryHigh       A-     Jan           Admit        Female
## 4   VeryHigh        A     Nov           Admit        Female
## 5   VeryHigh        A     Nov           Admit        Female
## 6   VeryHigh        A     Nov           Admit        Female
## 7   VeryHigh        A     Nov           Admit        Female
## 8     Medium        A     Nov           Admit        Female
## 9   VeryHigh        A     Nov           Admit        Female
## 10  VeryHigh        A     Jan           Admit        Female
## 11  VeryHigh        A     Nov           Admit        Female
## 12  VeryHigh        A     Nov           Admit        Female
## 13  VeryHigh        A     Nov           Admit        Female
## 14  VeryHigh       A+     Nov           Admit        Female
## 15  VeryHigh       A+     Dec           Admit        Female
## 16  VeryHigh       A+     Nov           Admit        Female
## 17  VeryHigh       A+     Dec           Admit        Female
## 18  VeryHigh        A     Jan           Admit        Female
## 19    Medium     C_B-     Jan         Decline        Female
## 20    Medium     C_B-     Jan         Decline        Female
## 21    Medium     C_B-     Jan         Decline        Female
## 22    Medium     C_B-     Jan         Decline        Female
## 23    Medium        B Feb_Mar         Decline        Female
## 24    Medium        B     Dec         Decline        Female
## 25    Medium        B Feb_Mar         Decline        Female
## 26    Medium        B Feb_Mar         Decline        Female
## 27    Medium        B     Dec         Decline        Female
## 28    Medium       A-     Nov         Decline        Female
## 29    Medium       A-     Jan         Decline        Female
## 30    Medium       A- Feb_Mar         Decline        Female
## 31    Medium       A-     Dec         Decline        Female
## 32    Medium        A     Jan         Decline        Female
## 33    Medium        A     Dec         Decline        Female
## 34      High        A     Nov         Decline        Female
## 35      High       A+     Jan         Decline        Female
## 36      High        B     Dec        Waitlist        Female
## 37      High        B Feb_Mar        Waitlist        Female
## 38      High       A-     Dec        Waitlist        Female
## 39      High       A-     Dec        Waitlist        Female
## 40      High       A-     Jan        Waitlist        Female
## 41      High       A- Feb_Mar        Waitlist        Female
## 42      High       A-     Jan        Waitlist        Female
## 43      High       A-     Dec        Waitlist        Female
## 44      High       A- Feb_Mar        Waitlist        Female
## 45      High       A-     Dec        Waitlist        Female
## 46      High       A-     Jan        Waitlist        Female
## 47      High       A-     Jan        Waitlist        Female
## 48      High       A-     Nov        Waitlist        Female
## 49      High       A-     Nov        Waitlist        Female
## 50      High       A-     Dec        Waitlist        Female
## 51      High       A-     Dec        Waitlist        Female
## 52      High       A-     Dec        Waitlist        Female
## 53      High       A-     Nov        Waitlist        Female
## 54      High       A-     Nov        Waitlist        Female
## 55      High       A-     Nov        Waitlist        Female
## 56  VeryHigh       A-     Jan           Admit          Male
## 57  VeryHigh        A     Nov           Admit          Male
## 58  VeryHigh        A     Dec           Admit          Male
## 59  VeryHigh        A     Dec           Admit          Male
## 60  VeryHigh        A     Nov           Admit          Male
## 61  VeryHigh        A     Nov           Admit          Male
## 62  VeryHigh        A     Nov           Admit          Male
## 63  VeryHigh        A     Dec           Admit          Male
## 64  VeryHigh        A     Nov           Admit          Male
## 65  VeryHigh       A+     Nov           Admit          Male
## 66  VeryHigh       A+     Nov           Admit          Male
## 67  VeryHigh       A+     Nov           Admit          Male
## 68  VeryHigh       A+     Nov           Admit          Male
## 69  VeryHigh       A+     Nov           Admit          Male
## 70  VeryHigh        A     Dec           Admit          Male
## 71    Medium     C_B- Feb_Mar         Decline          Male
## 72    Medium     C_B- Feb_Mar         Decline          Male
## 73    Medium     C_B- Feb_Mar         Decline           Male
## 74    Medium     C_B-     Jan         Decline           Male
## 75    Medium     C_B- Feb_Mar         Decline           Male
## 76    Medium        B Feb_Mar         Decline           Male
## 77    Medium        B     Jan         Decline           Male
## 78    Medium        B     Dec         Decline           Male
## 79      High       A-     Jan         Decline           Male
## 80    Medium        A     Dec         Decline           Male
## 81      High        B     Jan        Waitlist           Male
## 82      High       A-     Dec        Waitlist           Male
## 83      High       A- Feb_Mar        Waitlist           Male
## 84      High       A-     Dec        Waitlist           Male
## 85      High       A-     Nov        Waitlist           Male
##    MyData$VolunteerLevel
## 1                      1
## 2                      0
## 3                      0
## 4                      2
## 5                      2
## 6                      1
## 7                      2
## 8                      5
## 9                      0
## 10                     4
## 11                     2
## 12                     2
## 13                     5
## 14                     4
## 15                     1
## 16                     3
## 17                     0
## 18                     3
## 19                     1
## 20                     4
## 21                     1
## 22                     2
## 23                     0
## 24                     5
## 25                     5
## 26                     1
## 27                     2
## 28                     3
## 29                     0
## 30                     4
## 31                     4
## 32                     5
## 33                     4
## 34                     2
## 35                     1
## 36                     5
## 37                     4
## 38                     0
## 39                     1
## 40                     2
## 41                    0
## 42                    5
## 43                    5
## 44                    2
## 45                    4
## 46                    4
## 47                    2
## 48                    1
## 49                    2
## 50                    1
## 51                    1
## 52                    1
## 53                    3
## 54                    0
## 55                    5
## 56                    1
## 57                    1
## 58                    2
## 59                    3
## 60                    0
## 61                    2
## 62                    4
## 63                    4
## 64                    3
## 65                    5
## 66                    5
## 67                    0
## 68                    3
## 69                    4
## 70                    1
## 71                    5
## 72                    4
## 73                    1
## 74                    1
## 75                    0
## 76                    3
## 77                    1
## 78                    4
## 79                    3
## 80                    4
## 81                    4
## 82                    3
## 83                    3
## 84                    3
## 85                    3
```

```
MyARM_Data
```

```
##    TestScore GPALevel SubDate MyData$Decision MyData$Gender
## 1   VeryHigh       A-     Jan           Admit        Female
## 2   VeryHigh       A-     Jan           Admit        Female
## 3   VeryHigh       A-     Jan           Admit        Female
## 4   VeryHigh        A     Nov           Admit        Female
## 5   VeryHigh        A     Nov           Admit        Female
## 6   VeryHigh        A     Nov           Admit        Female
## 7   VeryHigh        A     Nov           Admit        Female
## 8     Medium        A     Nov           Admit        Female
## 9   VeryHigh        A     Nov           Admit        Female
## 10  VeryHigh        A     Jan           Admit        Female
## 11  VeryHigh        A     Nov           Admit        Female
## 12  VeryHigh        A     Nov           Admit        Female
## 13  VeryHigh        A     Nov           Admit        Female
## 14  VeryHigh       A+     Nov           Admit        Female
## 15  VeryHigh       A+     Dec           Admit        Female
## 16  VeryHigh       A+     Nov           Admit        Female
## 17  VeryHigh       A+     Dec           Admit        Female
## 18  VeryHigh        A     Jan           Admit        Female
## 19    Medium     C_B-     Jan         Decline        Female
## 20    Medium     C_B-     Jan         Decline        Female
## 21    Medium     C_B-     Jan         Decline        Female
## 22    Medium     C_B-     Jan         Decline        Female
## 23    Medium        B Feb_Mar         Decline        Female
## 24    Medium        B     Dec         Decline        Female
## 25    Medium        B Feb_Mar         Decline        Female
## 26    Medium        B Feb_Mar         Decline        Female
## 27    Medium        B     Dec         Decline        Female
## 28    Medium       A-     Nov         Decline        Female
## 29    Medium       A-     Jan         Decline        Female
## 30    Medium       A- Feb_Mar         Decline        Female
## 31    Medium       A-     Dec         Decline        Female
## 32    Medium        A     Jan         Decline        Female
## 33    Medium        A     Dec         Decline        Female
## 34      High        A     Nov         Decline        Female
## 35      High       A+     Jan         Decline        Female
## 36      High        B     Dec        Waitlist        Female
## 37      High        B Feb_Mar        Waitlist        Female
## 38      High       A-     Dec        Waitlist        Female
## 39      High       A-     Dec        Waitlist        Female
## 40      High       A-     Jan        Waitlist        Female
## 41      High       A- Feb_Mar        Waitlist        Female
## 42      High       A-     Jan        Waitlist        Female
## 43      High       A-     Dec        Waitlist        Female
## 44      High       A- Feb_Mar        Waitlist        Female
## 45      High       A-     Dec        Waitlist        Female
## 46      High       A-     Jan        Waitlist        Female
## 47      High       A-     Jan        Waitlist        Female
## 48      High       A-     Nov        Waitlist        Female
## 49      High       A-     Nov        Waitlist        Female
## 50      High       A-     Dec        Waitlist        Female
## 51      High       A-     Dec        Waitlist        Female
## 52      High       A-     Dec        Waitlist        Female
## 53      High       A-     Nov        Waitlist        Female
## 54      High       A-     Nov        Waitlist        Female
## 55      High       A-     Nov        Waitlist        Female
## 56  VeryHigh       A-     Jan           Admit          Male
## 57  VeryHigh        A     Nov           Admit          Male
## 58  VeryHigh        A     Dec           Admit          Male
## 59  VeryHigh        A     Dec           Admit          Male
## 60  VeryHigh        A     Nov           Admit          Male
## 61  VeryHigh        A     Nov           Admit          Male
## 62  VeryHigh        A     Nov           Admit          Male
## 63  VeryHigh        A     Dec           Admit          Male
## 64  VeryHigh        A     Nov           Admit          Male
## 65  VeryHigh       A+     Nov           Admit          Male
## 66  VeryHigh       A+     Nov           Admit          Male
## 67  VeryHigh       A+     Nov           Admit          Male
## 68  VeryHigh       A+     Nov           Admit          Male
## 69  VeryHigh       A+     Nov           Admit          Male
## 70  VeryHigh        A     Dec           Admit          Male
## 71    Medium     C_B- Feb_Mar         Decline          Male
## 72    Medium     C_B- Feb_Mar         Decline          Male
## 73    Medium     C_B- Feb_Mar         Decline          Male
## 74    Medium     C_B-     Jan         Decline          Male
## 75    Medium     C_B- Feb_Mar         Decline          Male
## 76    Medium        B Feb_Mar         Decline          Male
## 77    Medium        B     Jan         Decline          Male
## 78    Medium        B     Dec         Decline          Male
## 79      High       A-     Jan         Decline          Male
## 80    Medium        A     Dec         Decline          Male
## 81      High        B     Jan        Waitlist          Male
## 82      High       A-     Dec        Waitlist          Male
## 83      High       A- Feb_Mar        Waitlist          Male
## 84      High       A-     Dec        Waitlist          Male
## 85      High       A-     Nov        Waitlist          Male
##    MyData$VolunteerLevel
## 1                      1
## 2                      0
## 3                      0
## 4                      2
## 5                      2
## 6                      1
## 7                      2
## 8                      5
## 9                      0
## 10                     4
## 11                     2
## 12                     2
## 13                     5
## 14                     4
## 15                     1
## 16                     3
## 17                     0
## 18                     3
## 19                     1
## 20                     4
## 21                     1
## 22                     2
## 23                     0
## 24                     5
## 25                     5
## 26                     1
## 27                     2
## 28                        3
## 29                        0
## 30                        4
## 31                        4
## 32                        5
## 33                        4
## 34                        2
## 35                        1
## 36                        5
## 37                        4
## 38                        0
## 39                        1
## 40                        2
## 41                        0
## 42                        5
## 43                        5
## 44                        2
## 45                        4
## 46                        4
## 47                        2
## 48                        1
## 49                        2
## 50                        1
## 51                        1
## 52                        1
## 53                        3
## 54                        0
## 55                        5
## 56                        1
## 57                        1
## 58                        2
## 59                        3
## 60                        0
## 61                        2
## 62                        4
## 63                        4
## 64                        3
## 65                        5
## 66                        5
## 67                        0
## 68                        3
## 69                        4
## 70                        1
## 71                        5
## 72                        4
## 73                        1
## 74                        1
## 75                        0
## 76                        3
## 77                        1
## 78                        4
## 79                        3
## 80                        4
## 81                        4
## 82                     3
## 83                     3
## 84                     3
## 85                     3
```

```
## OK! Now we have a dataset such that
##
## EACH ROW IS A TRANSACTION containing meaningful words/labels

## Apply ARM
library(arules)
```

```
## Warning: package 'arules' was built under R version 3.5.3

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

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

```
library(arulesViz)
```

```
## Warning: package 'arulesViz' was built under R version 3.5.3

## Loading required package: grid
```

```
## IF  ERROR - use detach and then install the library
## detach("package:arulesViz", unload=TRUE)
## detach("package:arules", unload=TRUE)
## then - run
## library(arules)
## library(arulesViz)
##again

MY_rules <- arules::apriori(MyARM_Data,
                 parameter = list(supp = 0.25, conf = 0.25,
                                  target = "rules", minlen=2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.25    0.1    1 none FALSE            TRUE       5    0.25      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 21
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[23 item(s), 85 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [17 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(MY_rules[1:10])
```

```
##      lhs                           rhs                          support
## [1]  {MyData$Decision=Waitlist} => {TestScore=High}             0.2941176
## [2]  {TestScore=High}           => {MyData$Decision=Waitlist} 0.2941176
## [3]  {MyData$Decision=Waitlist} => {GPALevel=A-}                0.2588235
## [4]  {GPALevel=A-}              => {MyData$Decision=Waitlist} 0.2588235
## [5]  {TestScore=Medium}         => {MyData$Decision=Decline}  0.2823529
## [6]  {MyData$Decision=Decline}  => {TestScore=Medium}           0.2823529
## [7]  {TestScore=High}           => {GPALevel=A-}                0.2705882
## [8]  {GPALevel=A-}              => {TestScore=High}             0.2705882
## [9]  {TestScore=High}           => {MyData$Gender=Female}       0.2588235
## [10] {MyData$Gender=Female}     => {TestScore=High}             0.2588235
##      confidence lift     count
## [1]  1.0000000  3.035714 25
## [2]  0.8928571  3.035714 25
## [3]  0.8800000  2.412903 22
## [4]  0.7096774  2.412903 22
## [5]  0.9600000  3.022222 24
## [6]  0.8888889  3.022222 24
## [7]  0.8214286  2.252304 23
## [8]  0.7419355  2.252304 23
## [9]  0.7857143  1.214286 22
## [10] 0.4000000  1.214286 22
```

```
SortedRules_by_conf <- sort(MY_rules, by="confidence", decreasing=TRUE)
inspect(SortedRules_by_conf[1:10])
```

```
##      lhs                           rhs                            support confidence     lift count
## [1]  {MyData$Decision=Waitlist} => {TestScore=High}             0.2941176  1.0000000 3.035714    25
## [2]  {TestScore=VeryHigh}       => {MyData$Decision=Admit}      0.3764706  1.0000000 2.575758    32
## [3]  {GPALevel=A-,
##       MyData$Decision=Waitlist} => {TestScore=High}             0.2588235  1.0000000 3.035714    22
## [4]  {MyData$Decision=Admit}    => {TestScore=VeryHigh}         0.3764706  0.9696970 2.575758    32
## [5]  {TestScore=Medium}         => {MyData$Decision=Decline}  0.2823529  0.9600000 3.022222    24
## [6]  {TestScore=High,
##       GPALevel=A-}              => {MyData$Decision=Waitlist} 0.2588235  0.9565217 3.252174    22
## [7]  {TestScore=High}           => {MyData$Decision=Waitlist} 0.2941176  0.8928571 3.035714    25
## [8]  {MyData$Decision=Decline}  => {TestScore=Medium}         0.2823529  0.8888889 3.022222    24
## [9]  {MyData$Decision=Waitlist} => {GPALevel=A-}              0.2588235  0.8800000 2.412903    22
## [10] {TestScore=High,
##       MyData$Decision=Waitlist} => {GPALevel=A-}              0.2588235  0.8800000 2.412903    22
```

```
SortedRules_by_sup <- sort(MY_rules, by="support", decreasing=TRUE)
inspect(SortedRules_by_sup[1:10])
```

```
##      lhs                           rhs                        support
## [1]  {TestScore=VeryHigh}       => {MyData$Decision=Admit}    0.3764706
## [2]  {MyData$Decision=Admit}    => {TestScore=VeryHigh}       0.3764706
## [3]  {MyData$Decision=Waitlist} => {TestScore=High}           0.2941176
## [4]  {TestScore=High}           => {MyData$Decision=Waitlist} 0.2941176
## [5]  {GPALevel=A-}              => {MyData$Gender=Female}     0.2941176
## [6]  {MyData$Gender=Female}     => {GPALevel=A-}              0.2941176
## [7]  {TestScore=Medium}         => {MyData$Decision=Decline}  0.2823529
## [8]  {MyData$Decision=Decline}  => {TestScore=Medium}         0.2823529
## [9]  {TestScore=High}           => {GPALevel=A-}              0.2705882
## [10] {GPALevel=A-}              => {TestScore=High}           0.2705882
##      confidence lift     count
## [1]  1.0000000  2.575758 32
## [2]  0.9696970  2.575758 32
## [3]  1.0000000  3.035714 25
## [4]  0.8928571  3.035714 25
## [5]  0.8064516  1.246334 25
## [6]  0.4545455  1.246334 25
## [7]  0.9600000  3.022222 24
## [8]  0.8888889  3.022222 24
## [9]  0.8214286  2.252304 23
## [10] 0.7419355  2.252304 23
```

```
####### Visualize the results
## Uses arulesViz
plot (SortedRules_by_conf,
      method="graph",
      engine='interactive',
      shading="confidence")

plot (SortedRules_by_sup,
      method="graph",
      engine='interactive',
      shading="confidence")



######################################################################
##
##               Supervised Methods: Decision Trees
##
#########################################################################
```

```r
library(e1071)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

```
## Loading required package: lattice
```

```r
library(rpart)  ## For DT
library(rattle)
```

```
## Loading required package: tibble
```

```
## Warning: package 'tibble' was built under R version 3.5.3
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.3
```

```r
library(RColorBrewer)
library(Cairo)
```

```
## Warning: package 'Cairo' was built under R version 3.5.3
```

```r
## To perform supervised methods - we need to format the data
## and we need to create Training and Testing sets from the dataset.

head(MyData)
```

```
##   Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1    Admit Female 2020-01-11    Florida 3.54     0.7       965           94
## 2    Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3    Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4    Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5    Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6    Admit Female 2019-11-03 California 3.66     0.9       956           89
##   VolunteerLevel
## 1              1
## 2              0
## 3              0
## 4              2
## 5              2
## 6              1
```

```r
## Make sure all types are correct and Decision is type: FACTOR
## Decision should have 3 levels - Gender 2 levels - etc.
str(MyData)
```

```
## 'data.frame':    85 obs. of  9 variables:
##  $ Decision      : Factor w/ 3 levels "Admit","Decline",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender        : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DateSub       : Date, format: "2020-01-11" "2020-01-11" ...
##  $ State         : Factor w/ 11 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
##  $ GPA           : num  3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
##  $ WorkExp       : num  0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
##  $ TestScore     : int  965 962 969 969 967 956 969 799 969 969 ...
##  $ WritingScore  : int  94 97 93 97 94 89 94 97 93 99 ...
##  $ VolunteerLevel: Factor w/ 6 levels "0","1","2","3",..: 2 1 1 3 3 2 3 6 1 5 ...
```

```r
## double-check for NAs
## Note that we already cleaned that data above - so will not repeat here
(sum(is.na(MyData)))
```

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

```r
## -----------------------------
## Create the Train and Test sets
## ----------------------------------------

## First - remove the date column from the dataset
names(MyData)
```

```
## [1] "Decision"       "Gender"         "DateSub"        "State"
## [5] "GPA"            "WorkExp"        "TestScore"      "WritingScore"
## [9] "VolunteerLevel"
```

```r
head(MyData2<-MyData[,-3])  ## -3 because this is the date column
```

```
##   Decision Gender      State  GPA WorkExp TestScore WritingScore VolunteerLevel
## 1    Admit Female    Florida 3.54     0.7       965           94              1
## 2    Admit Female    Florida 3.55     0.0       962           97              0
## 3    Admit Female   Colorado 3.59     1.7       969           93              0
## 4    Admit Female   Colorado 3.60     0.9       969           97              2
## 5    Admit Female   Colorado 3.60     1.2       967           94              2
## 6    Admit Female California 3.66     0.9       956           89              1
```

```r
nrow(MyData2)
```

```
## [1] 85
```

```r
set.seed(1234)
(MySample <- sample(nrow(MyData2),nrow(MyData2)*.3))
```

```
##  [1] 10 53 51 52 70 82  1 19 80 40 84 41 21 67 73 59 20 78 13 16 71 69 11  3 14
```

```r
Testing_Set <- MyData2[MySample,]
Training_Set <- MyData2[-MySample,]

head(Testing_Set)
```

```
##    Decision Gender      State  GPA WorkExp TestScore WritingScore
## 10    Admit Female    Florida 3.77     1.4       969           99
## 54 Waitlist Female    Florida 3.56     1.3       869           94
## 52 Waitlist Female    Florida 3.55     2.0       853           94
## 53 Waitlist Female    Georgia 3.56     1.0       866           89
## 72    Admit   Male   Colorado 3.80     0.8       969           93
## 84 Waitlist   Male California 3.42     0.7       869           94
##    VolunteerLevel
## 10              4
## 54              3
## 52              1
## 53              1
## 72              1
## 84              3
```

```r
head(Training_Set)
```

```
##   Decision Gender      State  GPA WorkExp TestScore WritingScore VolunteerLevel
## 2    Admit Female    Florida 3.55     0.0       962           97              0
## 4    Admit Female   Colorado 3.60     0.9       969           97              2
## 5    Admit Female   Colorado 3.60     1.2       967           94              2
## 6    Admit Female California 3.66     0.9       956           89              1
## 7    Admit Female California 3.70     1.2       969           94              2
## 8    Admit Female California 3.70     2.7       799           97              5
```

```r
## Check the label balance! If its not excellent - re-sample
## If you set a seed each time, you can keep the one you like.
## Testing Set
TestG<-ggplot(Testing_Set) +
  geom_bar(aes(x = Decision, y = stat(count), fill = Decision))+
  theme(axis.text.x=element_text(angle=90,hjust=1))+
  geom_text(stat='count',aes(Decision, label=..count..),vjust=2)

TrainG<-ggplot(Training_Set) +
  geom_bar(aes(x = Decision, y = stat(count), fill = Decision))+
  theme(axis.text.x=element_text(angle=90,hjust=1))+
  geom_text(stat='count',aes(Decision, label=..count..),vjust=2)


#grid.arrange(TestG, TrainG, nrow = 2)

##---------------------------------------------------------
## HUGE !!!!!!!!!!!
##
## Now- remove the labels from the Test Data and Keep them
##----------------------------------------------------------
head(Testing_Set)
```

```
##    Decision Gender      State  GPA WorkExp TestScore WritingScore
## 10    Admit Female    Florida 3.77     1.4       969           99
## 54 Waitlist Female    Florida 3.56     1.3       869           94
## 52 Waitlist Female    Florida 3.55     2.0       853           94
## 53 Waitlist Female    Georgia 3.56     1.0       866           89
## 72    Admit   Male   Colorado 3.80     0.8       969           93
## 84 Waitlist   Male California 3.42     0.7       869           94
##    VolunteerLevel
## 10              4
## 54              3
## 52              1
## 53              1
## 72              1
## 84              3
```

```
MyTestLabels<-Testing_Set[,1]
head(MyTestLabels)
```

```
## [1] Admit    Waitlist Waitlist Waitlist Admit    Waitlist
## Levels: Admit Decline Waitlist
```

```
Testing_Set_No_Labels<-Testing_Set[,-1]
head(Testing_Set_No_Labels)
```

```
##    Gender      State  GPA WorkExp TestScore WritingScore VolunteerLevel
## 10 Female    Florida 3.77     1.4       969           99              4
## 54 Female    Florida 3.56     1.3       869           94              3
## 52 Female    Florida 3.55     2.0       853           94              1
## 53 Female    Georgia 3.56     1.0       866           89              1
## 72   Male   Colorado 3.80     0.8       969           93              1
## 84   Male California 3.42     0.7       869           94              3
```

```
###############################
## Train and Test the Tree
##
## Visualize each tree as you
## update.
##############################################

## Train the tree
## Recall that trees are created randomly by R
## Recall that there are an infinite number of
## possible trees.
##
## By making updates to the data - such as selecting
## specific columns, using feature generation,
## or using discretization - different trees can be
## created and visualized.

#################################################
##
MyTree1 <- rpart(Decision ~ ., data = Training_Set, method="class")
summary(MyTree1)
```

```
## Call:
## rpart(formula = Decision ~ ., data = Training_Set, method = "class")
##   n= 60
##
##          CP nsplit rel error    xerror       xstd
## 1 0.5000000      0 1.0000000 1.0789474 0.09482209
## 2 0.3947368      1 0.5000000 0.5526316 0.09722607
## 3 0.0100000      2 0.1052632 0.1052632 0.05084694
##
## Variable importance
##    TestScore          GPA WritingScore        State      WorkExp       Gender
##           44           28           11            8            7            4
##
## Node number 1: 60 observations,    complexity param=0.5
##   predicted class=Admit     expected loss=0.6333333  P(node) =1
##     class counts:    22    20    18
##    probabilities: 0.367 0.333 0.300
##   left son=2 (21 obs) right son=3 (39 obs)
##   Primary splits:
##       TestScore    < 927.5 to the right,   improve=19.456410, (0 missing)
##       GPA          < 3.59  to the right,   improve=13.422220, (0 missing)
##       WritingScore < 94.5  to the right,   improve= 6.376471, (0 missing)
##       State        splits as  RLLR--RRRRR, improve= 3.120255, (0 missing)
##       Gender       splits as  RL,          improve= 1.716667, (0 missing)
##   Surrogate splits:
##       GPA          < 3.59  to the right,   agree=0.883, adj=0.667, (0 split)
##       WritingScore < 94.5  to the right,   agree=0.767, adj=0.333, (0 split)
##       State        splits as  RLRR--RRRRR, agree=0.700, adj=0.143, (0 split)
##       Gender       splits as  RL,          agree=0.683, adj=0.095, (0 split)
##       WorkExp      < 0.35  to the left,    agree=0.683, adj=0.095, (0 split)
##
## Node number 2: 21 observations
##   predicted class=Admit     expected loss=0  P(node) =0.35
##     class counts:    21     0     0
##    probabilities: 1.000 0.000 0.000
##
## Node number 3: 39 observations,    complexity param=0.3947368
##   predicted class=Decline   expected loss=0.4871795  P(node) =0.65
##     class counts:     1    20    18
##    probabilities: 0.026 0.513 0.462
##   left son=6 (17 obs) right son=7 (22 obs)
##   Primary splits:
##       TestScore    < 794   to the left,    improve=13.592070, (0 missing)
##       GPA          < 3.38  to the left,    improve= 5.317664, (0 missing)
##       State        splits as  RRRL--RLRRL, improve= 3.283272, (0 missing)
##       WritingScore < 88.5  to the right,   improve= 3.240902, (0 missing)
##       WorkExp      < 1.15  to the left,    improve= 1.023160, (0 missing)
##   Surrogate splits:
##       GPA          < 3.38  to the left,    agree=0.821, adj=0.588, (0 split)
##       State        splits as  RRRR--RLRRL, agree=0.667, adj=0.235, (0 split)
##       WorkExp      < 1.15  to the left,    agree=0.667, adj=0.235, (0 split)
##       WritingScore < 91.5  to the right,   agree=0.615, adj=0.118, (0 split)
##       Gender       splits as  RL,          agree=0.590, adj=0.059, (0 split)
##
```

```
## Node number 6: 17 observations
##   predicted class=Decline   expected loss=0  P(node) =0.2833333
##     class counts:     0    17     0
##    probabilities: 0.000 1.000 0.000 
## 
## Node number 7: 22 observations
##   predicted class=Waitlist  expected loss=0.1818182  P(node) =0.3666667
##     class counts:     1     3    18
##    probabilities: 0.045 0.136 0.818
```

```
## What was the most important variable?  (TestScore)
## What was the second most important? (GPA)
## Which was least important? (Gender)

##--------------------Predictions...................
## Check your model on the Test data
## Notice you MUST use the Test set with NO LABELS

MyModelPrediction= predict(MyTree1,Testing_Set_No_Labels, type="class")
(MyResults <- data.frame(Predicted=MyModelPrediction,Actual=MyTestLabels))
```

```
##    Predicted   Actual
## 10     Admit    Admit
## 54  Waitlist Waitlist
## 52  Waitlist Waitlist
## 53  Waitlist Waitlist
## 72     Admit    Admit
## 84  Waitlist Waitlist
## 1      Admit    Admit
## 20   Decline  Decline
## 82  Waitlist  Decline
## 41  Waitlist Waitlist
## 86  Waitlist Waitlist
## 42  Waitlist Waitlist
## 22   Decline  Decline
## 69     Admit    Admit
## 75   Decline  Decline
## 61     Admit    Admit
## 21   Decline  Decline
## 80   Decline  Decline
## 13     Admit    Admit
## 16     Admit    Admit
## 73   Decline  Decline
## 71     Admit    Admit
## 11     Admit    Admit
## 3      Admit    Admit
## 14     Admit    Admit
```

```
## Basic Confusion Matrix
(MyTable<-table(MyModelPrediction,MyTestLabels))
```

```
##                   MyTestLabels
## MyModelPrediction Admit Decline Waitlist
##          Admit       11       0        0
##          Decline      0       6        0
##          Waitlist     0       1        7
```

```
str(MyTable)
```

```
##  'table' int [1:3, 1:3] 11 0 0 0 6 1 0 0 7
##  - attr(*, "dimnames")=List of 2
##   ..$ MyModelPrediction: chr [1:3] "Admit" "Decline" "Waitlist"
##   ..$ MyTestLabels     : chr [1:3] "Admit" "Decline" "Waitlist"
```

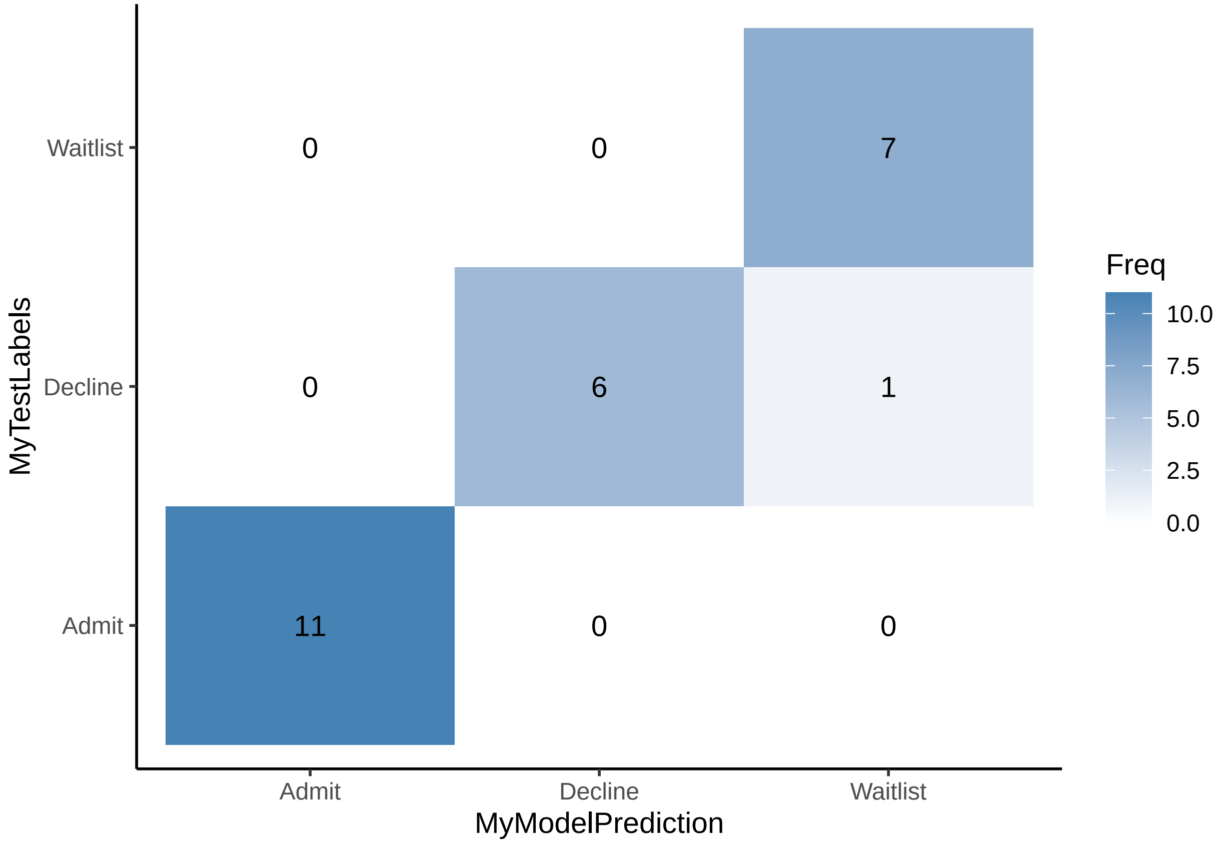
```
## Create a DF from the table to use in the heat map below...
(MyTable_DF<-as.data.frame(MyTable))
```

```
##   MyModelPrediction MyTestLabels Freq
## 1             Admit        Admit   11
## 2           Decline        Admit    0
## 3          Waitlist        Admit    0
## 4             Admit      Decline    0
## 5           Decline      Decline    6
## 6          Waitlist      Decline    1
## 7             Admit     Waitlist    0
## 8           Decline     Waitlist    0
## 9          Waitlist     Waitlist    7
```

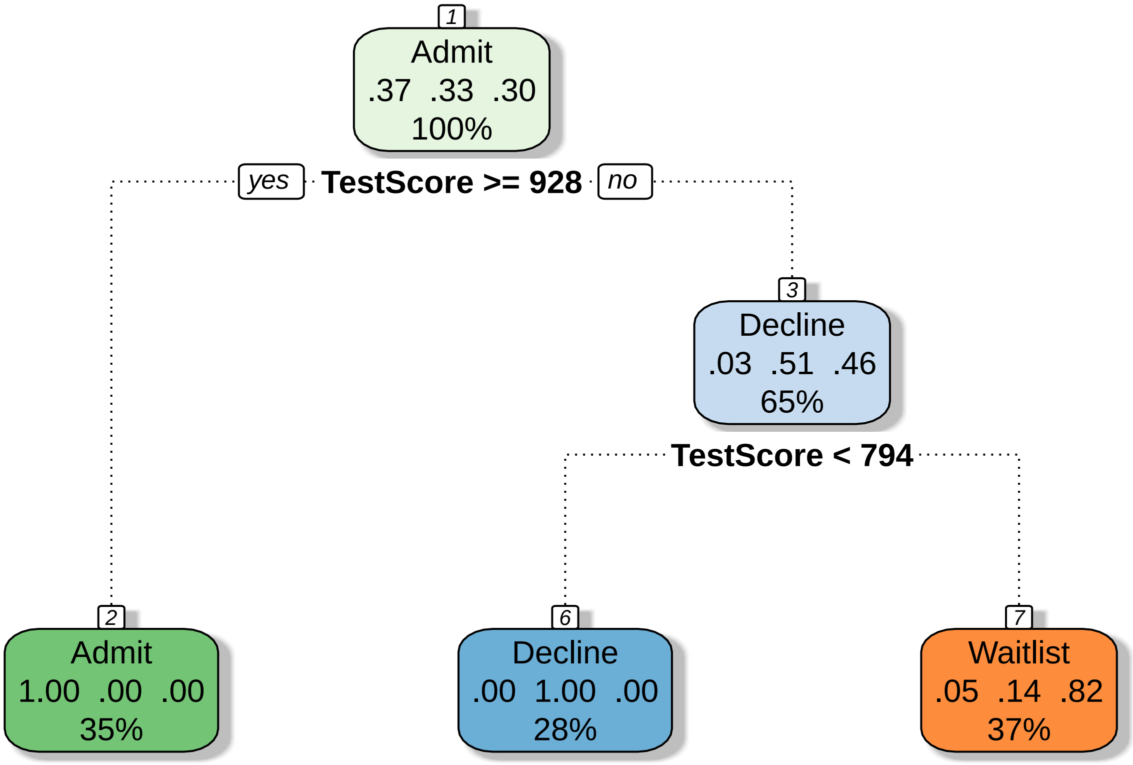
```
str(MyTable_DF)
```

```
## 'data.frame':    9 obs. of  3 variables:
##  $ MyModelPrediction: Factor w/ 3 levels "Admit","Decline",..: 1 2 3 1 2 3 1 2 3
##  $ MyTestLabels     : Factor w/ 3 levels "Admit","Decline",..: 1 1 1 2 2 2 3 3 3
##  $ Freq             : int  11 0 0 0 6 1 0 0 7
```

```
## Alternative - this offer sensitivity, specificity, accuracy, etc.
#(MyConf_Mat <- confusionMatrix(MyModelPrediction,MyTestLabels))
## To see the table:
#MyConf_Mat$table
#str(MyConf_Mat)

## Nicer Confusion Matrix - but only works on 2x2
## fourfoldplot(MyConf_Mat$table)

## Using ggplot heatmap to build a confusion matrix
ggplot(MyTable_DF, aes(x=MyModelPrediction, y=MyTestLabels, fill=Freq)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "steelblue")+
  geom_text(aes(label=Freq))
```

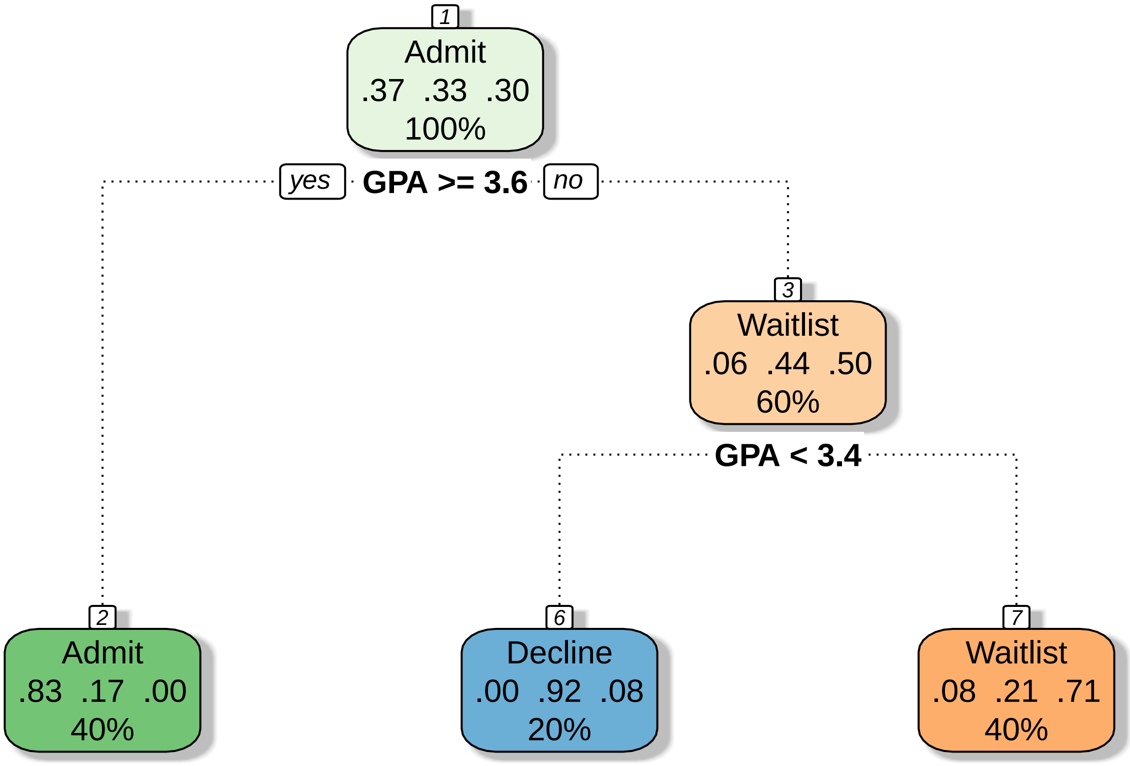


```
#################################
##
## Now - let's build some trees!
##
##########################################
fancyRpartPlot(MyTree1)
```



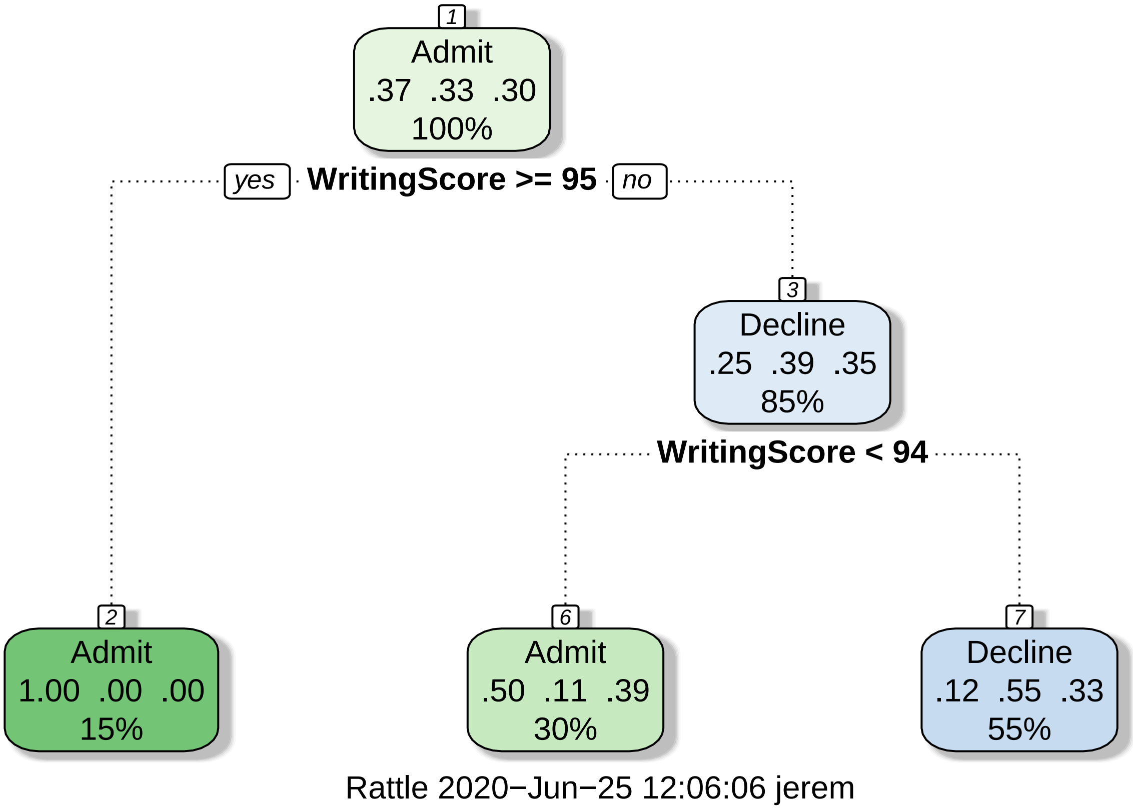
Rattle 2020−Jun−25 12:06:05 jerem

```
## Since TestScore is taking over - let's
## built a tree without it...

MyTree2 <- rpart(Decision ~ GPA+Gender, data = Training_Set, method="class")
#summary(MyTree2)
fancyRpartPlot(MyTree2)
```

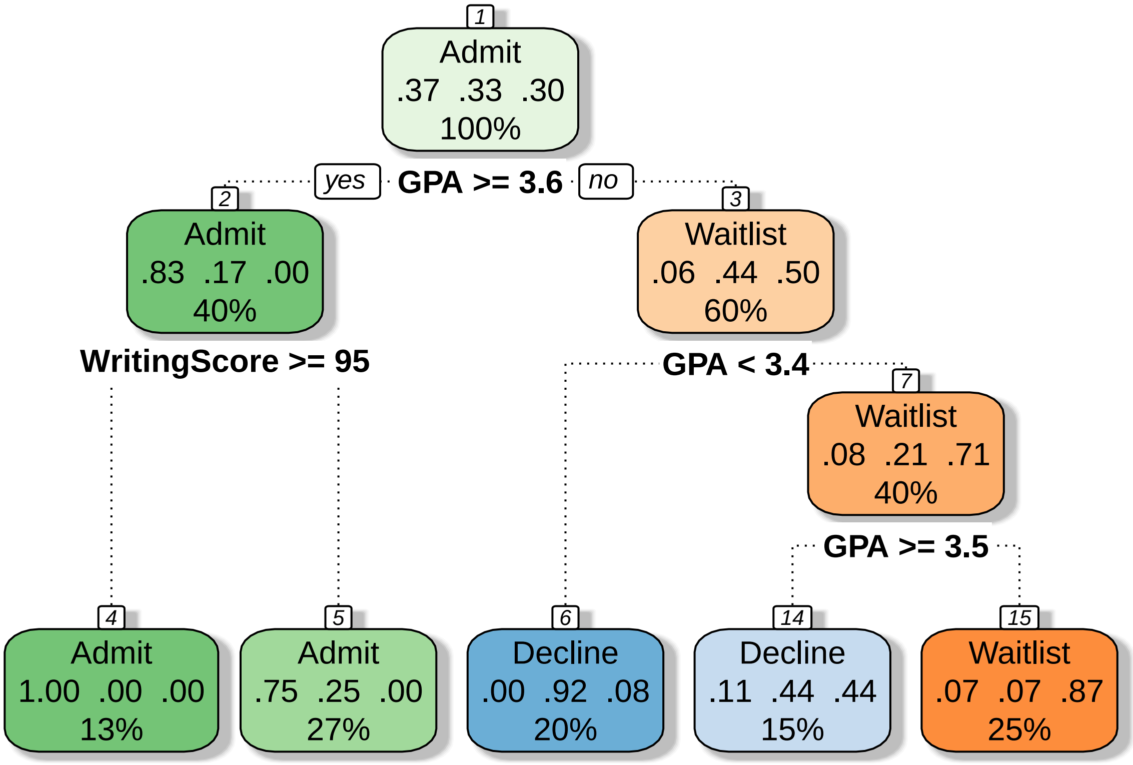


Rattle 2020−Jun−25 12:06:06 jerem

```
## How about WritingScore and VolunteerLevel?
MyTree3 <- rpart(Decision ~ WritingScore + VolunteerLevel,
                 data = Training_Set, method="class")
#summary(MyTree3)
fancyRpartPlot(MyTree3)
```



```r
##Setting cp to a negative amount
## ensures that the tree will be fully grown.

MyTree4 <- rpart(Decision ~ WritingScore +GPA,
                 data = Training_Set, method="class", cp=-1)
#summary(MyTree4)
fancyRpartPlot(MyTree4)
```



Rattle 2020−Jun−25 12:06:06 jerem

```
######################### There are so many options to try!
#-----------------------------------------
########## Information: Top Attributes ###
#-----------------------------------------
library(FSelector)
```

```
## Warning: package 'FSelector' was built under R version 3.5.3
```

```
(My_FSelector <- FSelector::information.gain(Decision ~ .,data=Training_Set))
```

```
##                attr_importance
## Gender              0.04357041
## State               0.24031487
## GPA                 0.54642868
## WorkExp             0.00000000
## TestScore           0.96561148
## WritingScore        0.17464957
## VolunteerLevel      0.02676032
```

```
####################################################
##
##   Random Forest
##
###################################################
```

```
##install.packages("randomForest")
## Save and restart if you need to - SAVE FIRST
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':
##
##     importance

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

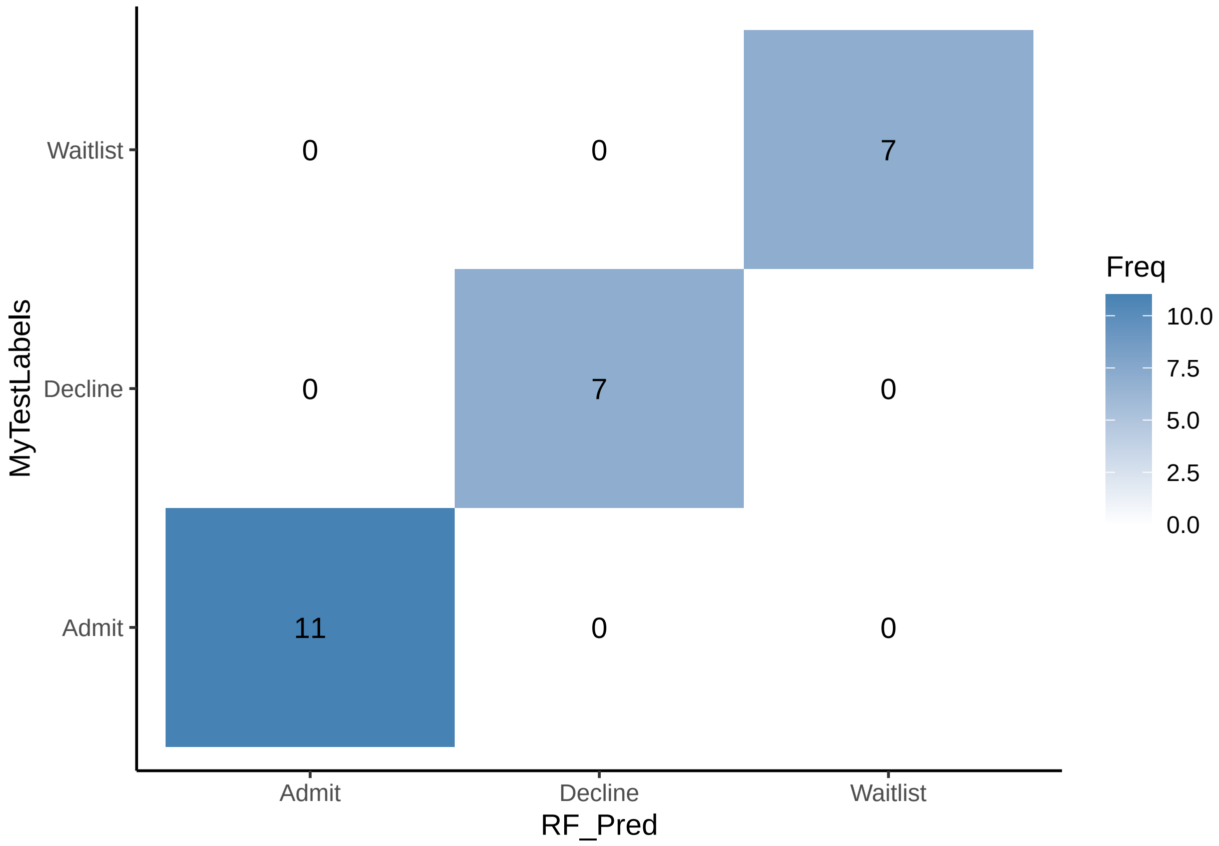
```
My_RF <- randomForest(Decision ~ .,data=Training_Set)

RF_Pred= predict(My_RF,Testing_Set_No_Labels, type="class")

## Basic Confusion Matrix
(My_RF_Table<-table(RF_Pred,MyTestLabels))
```

```
##            MyTestLabels
## RF_Pred     Admit Decline Waitlist
##   Admit        11       0        0
##   Decline       0       7        0
##   Waitlist      0       0        7
```

```
## Create a DF from the table to use in the heat map below...
(My_RF_Table_DF<-as.data.frame(My_RF_Table))
```

```
##    RF_Pred MyTestLabels Freq
## 1    Admit        Admit   11
## 2  Decline        Admit    0
## 3 Waitlist        Admit    0
## 4    Admit      Decline    0
## 5  Decline      Decline    7
## 6 Waitlist      Decline    0
## 7    Admit     Waitlist    0
## 8  Decline     Waitlist    0
## 9 Waitlist     Waitlist    7
```

```r
## Using ggplot heatmap to build a confusion matrix
ggplot(My_RF_Table_DF,
       aes(x=RF_Pred, y=MyTestLabels, fill=Freq)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "steelblue")+
  geom_text(aes(label=Freq))
```



```r
## Other options
My_RF$confusion
```

```
##          Admit Decline Waitlist class.error
## Admit       21       1        0  0.04545455
## Decline      1      17        2  0.15000000
## Waitlist     0       2       16  0.11111111
```

```r
My_RF$importance  ## Includes GINI
```

```
##                MeanDecreaseGini
## Gender                0.7985908
## State                 2.9856790
## GPA                   9.4942882
## WorkExp               2.2160473
## TestScore            17.7840469
## WritingScore          3.5354206
## VolunteerLevel        2.1077066
```

```r
##My_RF$forest - UGLY



## Using party ###############
library(party)
```

```
## Warning: package 'party' was built under R version 3.5.3

## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 3.5.3

##
## Attaching package: 'mvtnorm'

## The following object is masked from 'package:mclust':
##
##     dmvnorm

## Loading required package: modeltools

## Warning: package 'modeltools' was built under R version 3.5.3

## Loading required package: stats4

##
## Attaching package: 'modeltools'

## The following object is masked from 'package:arules':
##
##     info

## The following object is masked from 'package:plyr':
##
##     empty

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 3.5.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.5.3

##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 3.5.3
```

```
## Important ##
## Random Forest is RANDOM!
## So you will get different answers each time - AND -
## if you it does not run right - just run it again
##
CF_RF<-cforest(Decision ~ .,
        data=Training_Set,
        controls=cforest_control(mtry=2, mincriterion=0))


#CF_RF@data
## Note - the "@" is used with objects like $ is used with DFs

MyParty<-party:::prettytree(CF_RF@ensemble[[1]],
                            names(CF_RF@data@get("input")))
MyNewTree <- new("BinaryTree")
MyNewTree@tree <- MyParty
MyNewTree@data <- CF_RF@data
MyNewTree@responses <- CF_RF@responses
MyNewTree
```

```
##
##   Conditional inference tree with 0 terminal nodes
##
## Response:  Decision
## Inputs:  Gender, State, GPA, WorkExp, TestScore, WritingScore, VolunteerLevel
## Number of observations:  60
##
## 1) State == {}; criterion = 3.541, statistic = 3.541
##   2)*  weights = 0
## 1) State == {}
##   3) VolunteerLevel == {}; criterion = 2.103, statistic = 2.103
##     4)*  weights = 0
##   3) VolunteerLevel == {}
##     5) GPA <= 3.58; criterion = 3.756, statistic = 3.756
##       6) WorkExp <= 1.4; criterion = 0.255, statistic = 1.375
##         7)*  weights = 0
##       6) WorkExp > 1.4
##         8)*  weights = 0
##     5) GPA > 3.58
##       9)*  weights = 0
```

```
#plot(MyNewTree)
```

```
### caret has a good RF model tool as well
## "rf" is random forest
## This is nice as it offers Accuracy and Kappa
library(caret)
RF_caret <- caret::train(Decision~ ., method="rf", data=Training_Set)
RF_caret
```

```
## Random Forest
##
## 60 samples
##  7 predictor
##  3 classes: 'Admit', 'Decline', 'Waitlist'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 60, 60, 60, 60, 60, 60, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.7934005  0.6888944
##   11    0.9347746  0.8985698
##   20    0.9363234  0.9008351
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 20.
```

```
############# Using ggplot and caret to see more---------
# Save the variable importance values from our model
## object generated from caret.
(ImportantVariables<-varImp(RF_caret, scale = FALSE))
```

```
## rf variable importance
##
##                    Overall
## TestScore        33.029680
## GPA               4.936685
## StateFlorida      0.330318
## WritingScore      0.232500
## WorkExp           0.204180
## StateCalifornia   0.183372
## VolunteerLevel5   0.161262
## StateUtah         0.046756
## VolunteerLevel3   0.033234
## VolunteerLevel2   0.027600
## VolunteerLevel1   0.026796
## GenderMale        0.014073
## StateNew York     0.003544
## StateColorado     0.000000
## StateOregon       0.000000
## StateVirginia     0.000000
## VolunteerLevel4   0.000000
## StateVermont      0.000000
## StateGeorgia      0.000000
## Statemississippi  0.000000
```

```r
# Get the row names of the variable importance data
rownames(ImportantVariables$importance)
```

```
##  [1] "GenderMale"       "StateCalifornia"  "StateColorado"    "StateFlorida"
##  [5] "StateGeorgia"     "Statemississippi" "StateNew York"    "StateOregon"
##  [9] "StateUtah"        "StateVermont"     "StateVirginia"    "GPA"
## [13] "WorkExp"          "TestScore"        "WritingScore"     "VolunteerLevel1"
## [17] "VolunteerLevel2"  "VolunteerLevel3"  "VolunteerLevel4"  "VolunteerLevel5"
```

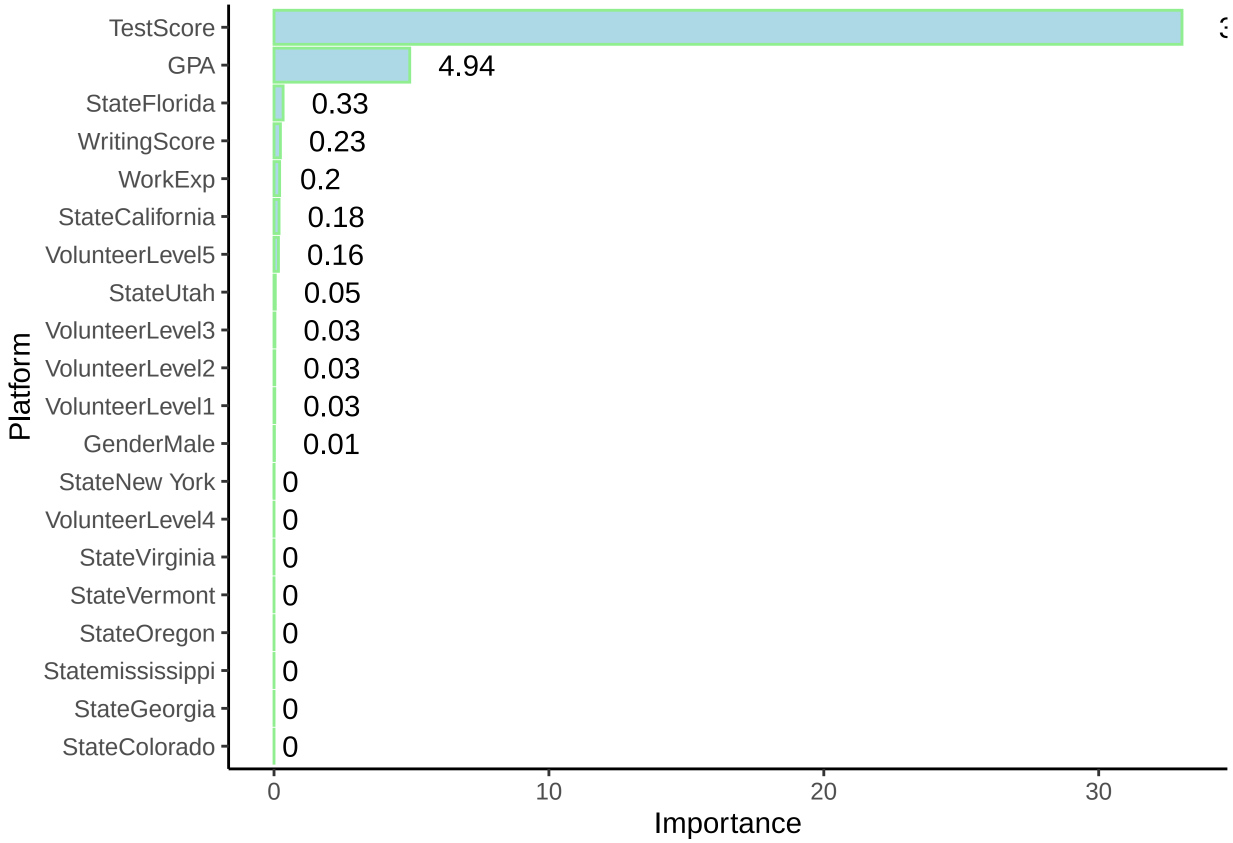
```r
# Convert the variable importance data into a dataframe
Import_DF <- data.frame(rownames(ImportantVariables$importance),
                        ImportantVariables$importance$Overall)
head(Import_DF)
```

```
##   rownames.ImportantVariables.importance. ImportantVariables.importance.Overall
## 1                              GenderMale                            0.01407273
## 2                         StateCalifornia                            0.18337218
## 3                           StateColorado                            0.00000000
## 4                            StateFlorida                            0.33031836
## 5                            StateGeorgia                            0.00000000
## 6                        Statemississippi                            0.00000000
```

```r
# Relabel the data
names(Import_DF)<-c('Platform', 'Importance')
# Order the data from greatest importance to least important
Import_DF <- transform(Import_DF,
                        Platform = reorder(Platform, Importance))

(Import_DF)
```

```
##            Platform   Importance
## 1        GenderMale  0.014072727
## 2   StateCalifornia  0.183372179
## 3     StateColorado  0.000000000
## 4      StateFlorida  0.330318356
## 5      StateGeorgia  0.000000000
## 6  Statemississippi  0.000000000
## 7     StateNew York  0.003544118
## 8       StateOregon  0.000000000
## 9         StateUtah  0.046755556
## 10     StateVermont  0.000000000
## 11    StateVirginia  0.000000000
## 12              GPA  4.936685064
## 13          WorkExp  0.204179818
## 14        TestScore 33.029679714
## 15     WritingScore  0.232500456
## 16  VolunteerLevel1  0.026796022
## 17  VolunteerLevel2  0.027600000
## 18  VolunteerLevel3  0.033234085
## 19  VolunteerLevel4  0.000000000
## 20  VolunteerLevel5  0.161261905
```

```r
# Plot the data with ggplot.
(MyPlot<-ggplot(data=Import_DF, aes(x=Platform, y=Importance)) +
  geom_bar(stat = 'identity',colour = "lightgreen",
           fill = "lightblue") +
  geom_text(aes(label=round(Importance,2), hjust=-.5))+
  coord_flip())
```



```r
####################################################
## SAving Plots
####################################################

## if you wish...
## setwd("/Users/SomeName/Desktop/etc")

## Options:
#ggsave("HorizBar.jpeg")
#ggsave(MyPlot, file="HorizBar.jpeg")
ggsave(MyPlot, file="HorizBar.jpeg", width=4, height=8)


#######################################################
##
##              Naive Bayes
##
#######################################################
```

```r
library(e1071)

NBStudentclassfier <- naiveBayes(Decision ~.,
                                 data=Training_Set,
                                 na.action = na.pass)

NBStudentClassifier_Prediction <- predict(NBStudentclassfier,
                                          Testing_Set_No_Labels)

## Basic Confusion Matrix
table(NBStudentClassifier_Prediction,MyTestLabels)
```

```
##                               MyTestLabels
## NBStudentClassifier_Prediction Admit Decline Waitlist
##                       Admit       11       0        0
##                       Decline      0       7        1
##                       Waitlist     0       0        6
```

```r
## This creates excellent output
## It is a good idea to think about how to BUILD
## a pretty figure with this output
NBStudentclassfier
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     Admit   Decline  Waitlist
## 0.3666667 0.3333333 0.3000000
##
## Conditional probabilities:
##           Gender
## Y             Female      Male
##   Admit    0.5000000 0.5000000
##   Decline  0.7000000 0.3000000
##   Waitlist 0.8333333 0.1666667
##
##           State
## Y              Alabama California   Colorado    Florida    Georgia mississippi
##   Admit    0.00000000 0.31818182 0.27272727 0.36363636 0.00000000  0.00000000
##   Decline  0.00000000 0.00000000 0.15000000 0.55000000 0.00000000  0.00000000
##   Waitlist 0.05555556 0.11111111 0.22222222 0.33333333 0.00000000  0.00000000
##           State
## Y             New York     Oregon       Utah    Vermont   Virginia
##   Admit    0.00000000 0.00000000 0.04545455 0.00000000 0.00000000
##   Decline  0.00000000 0.05000000 0.10000000 0.00000000 0.15000000
##   Waitlist 0.05555556 0.00000000 0.16666667 0.05555556 0.00000000
##
```

```
##           GPA
## Y               [,1]       [,2]
##   Admit    3.734091 0.11754063
##   Decline  3.367500 0.27343574
##   Waitlist 3.481667 0.07461667
##
##           WorkExp
## Y               [,1]      [,2]
##   Admit    1.595455 1.0934773
##   Decline  1.670000 0.8676041
##   Waitlist 1.950000 1.0262725
##
##           TestScore
## Y               [,1]      [,2]
##   Admit    958.5909 35.789942
##   Decline  785.8000 44.972038
##   Waitlist 866.1111  3.894021
##
##           WritingScore
## Y               [,1]     [,2]
##   Admit    93.54545 3.262007
##   Decline  93.40000 1.875044
##   Waitlist 90.94444 4.006938
##
##           VolunteerLevel
## Y                   0          1          2          3          4          5
##   Admit    0.18181818 0.18181818 0.27272727 0.13636364 0.09090909 0.13636364
##   Decline  0.15000000 0.20000000 0.15000000 0.15000000 0.20000000 0.15000000
##   Waitlist 0.11111111 0.16666667 0.16666667 0.11111111 0.22222222 0.22222222
```

```
## To the figure, you can also include other vis.

## !!!!!!!! Different libraries offer different vis options
## Above, I like the output that e1071 gives.
## But- below - I will use Caret and the vis options...


library(caret)
library(klaR)
```

```
## Warning: package 'klaR' was built under R version 3.5.3
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
## NB in caret
caret_NB = train(Training_Set[,-1], ## data and NOT label
                 Training_Set[,1],  ## the label ONLY
                 'nb',
                 trControl=trainControl(method='cv',number=100))
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```r
caret_NB
```

```
## Naive Bayes
##
## 60 samples
##  7 predictor
##  3 classes: 'Admit', 'Decline', 'Waitlist'
##
## No pre-processing
## Resampling: Cross-Validated (100 fold)
## Summary of sample sizes: 59, 59, 59, 59, 59, 59, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9166667  0
##    TRUE      0.9166667  0
```

```
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE and adjust
##  = 1.
```

```
Predict <- predict(caret_NB,Testing_Set )
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 8

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 17

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 21

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 23
```

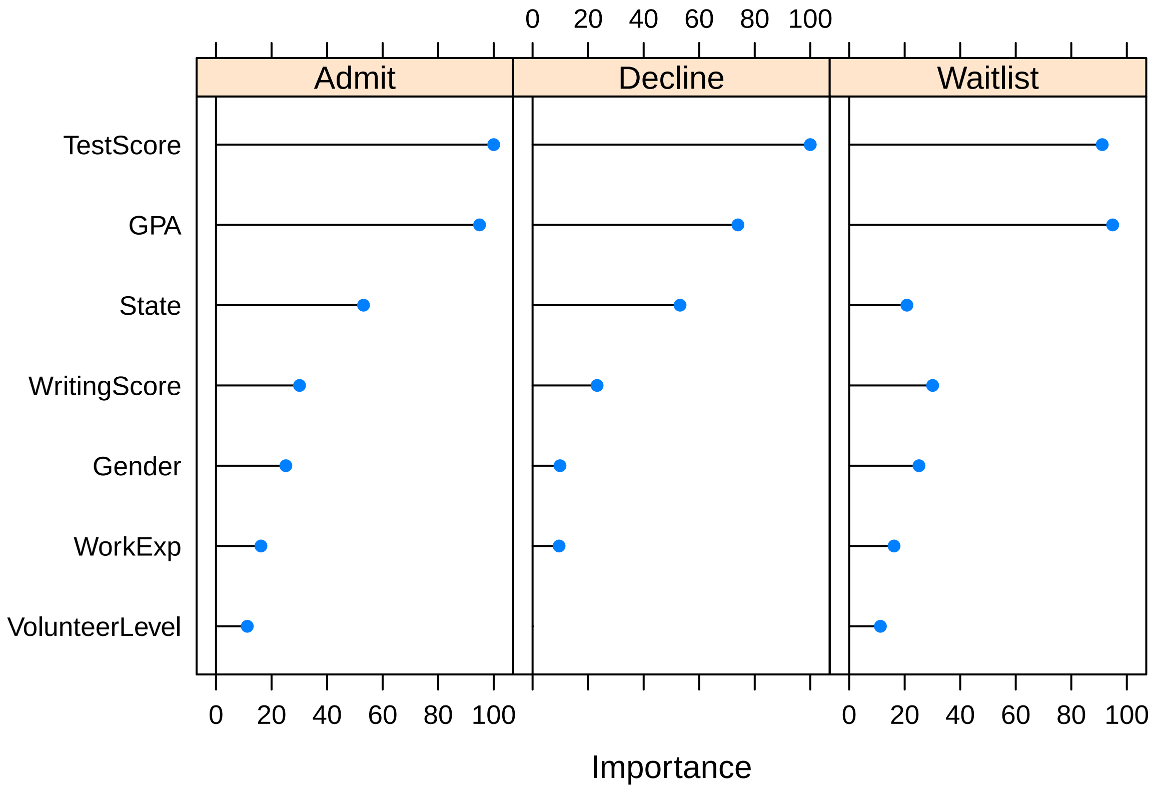
```
Predict
```

```
##  [1] Admit    Waitlist Decline  Waitlist Admit    Waitlist Admit    Decline
##  [9] Decline  Waitlist Waitlist Waitlist Decline  Admit    Decline  Admit
## [17] Decline  Decline  Admit    Admit    Decline  Admit    Admit    Admit
## [25] Admit
## Levels: Admit Decline Waitlist
```

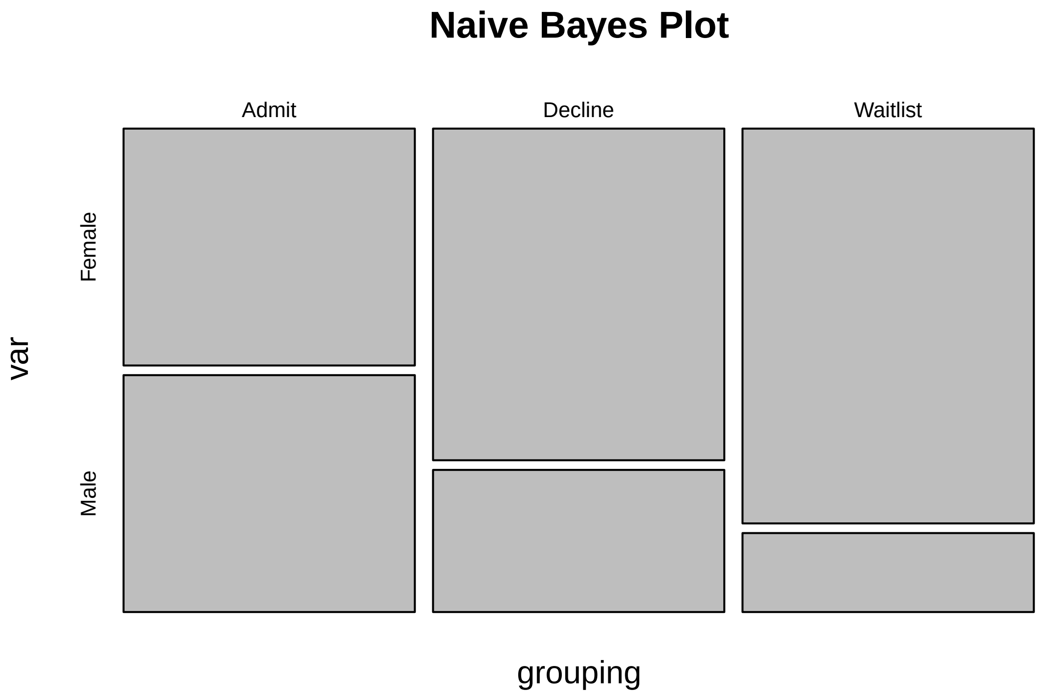
```
table(Predict,MyTestLabels)
```

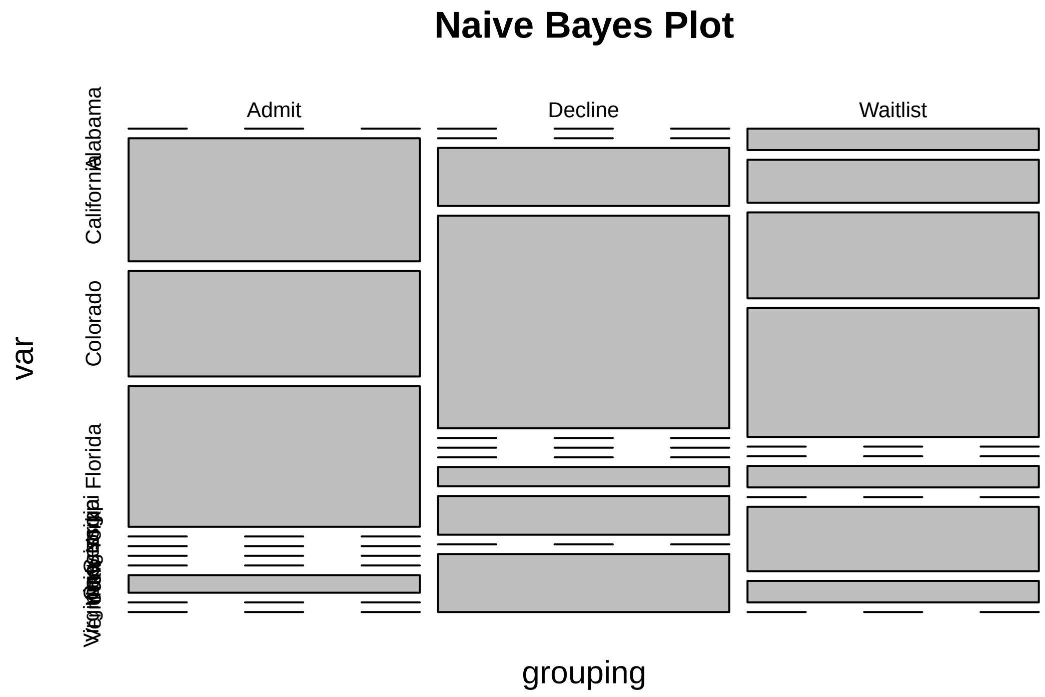
```
##           MyTestLabels
## Predict    Admit Decline Waitlist
##   Admit       11       0        0
##   Decline      0       7        1
##   Waitlist     0       0        6
```

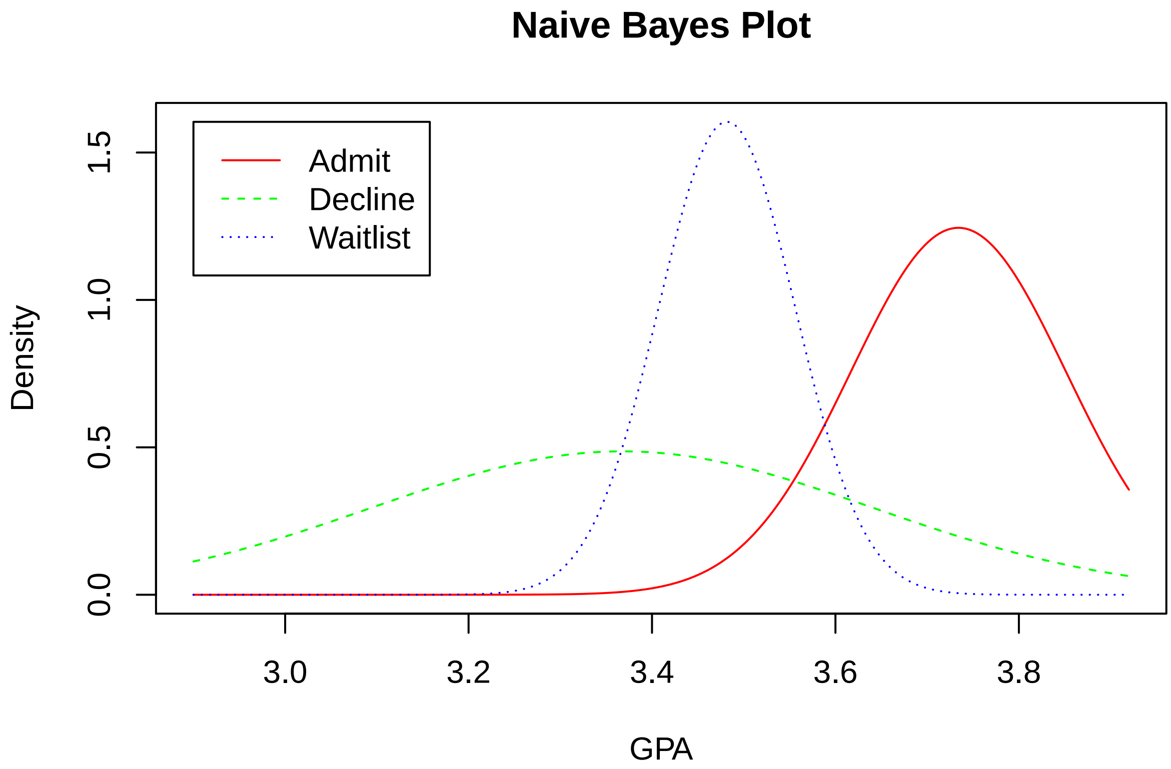
```
# Variable importance - this is a fun plot!
VarImp_NB <- caret::varImp(caret_NB)
plot(VarImp_NB)
```

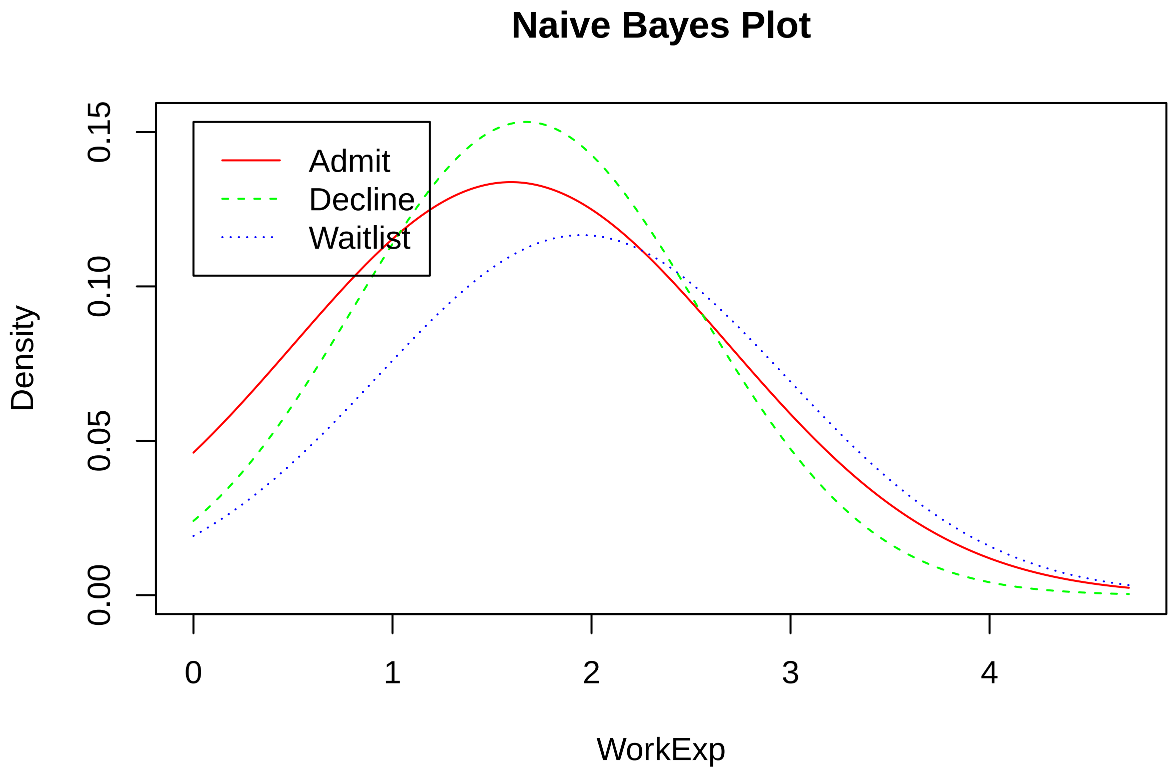


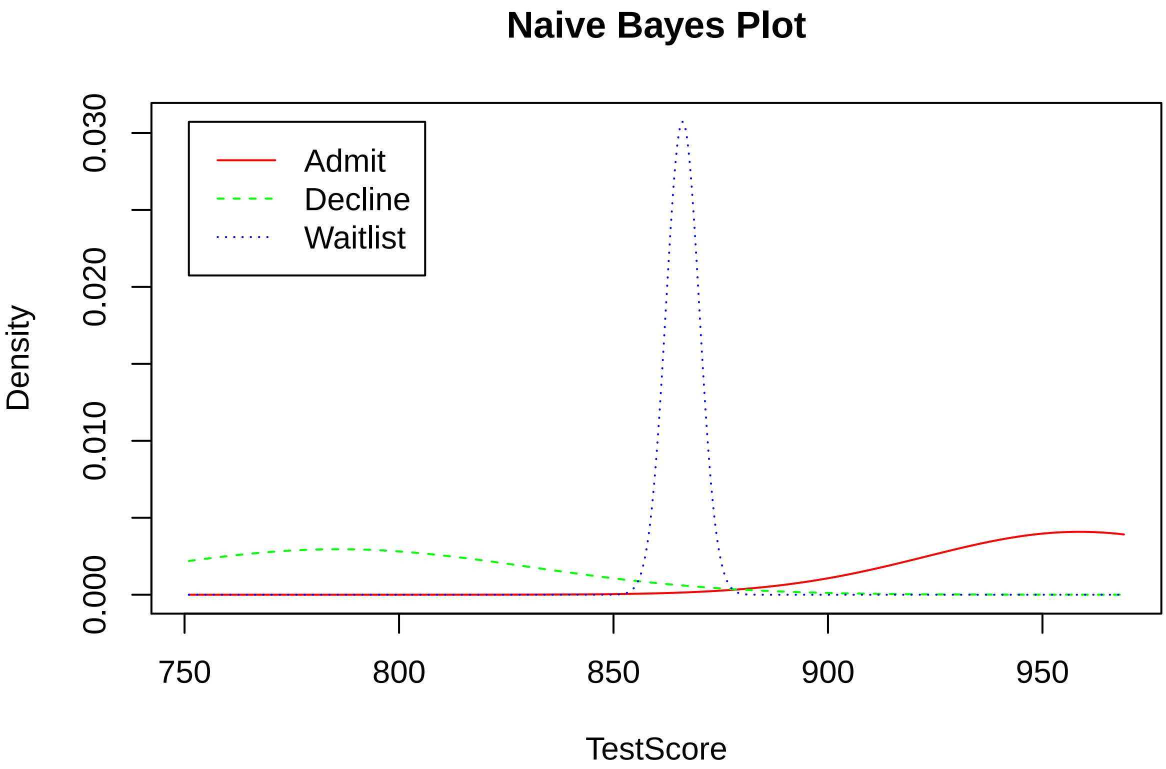
```
##
## Using klaR ---------------------

klaR_NB <- NaiveBayes(Decision ~ ., data = Training_Set)
plot(klaR_NB)
```

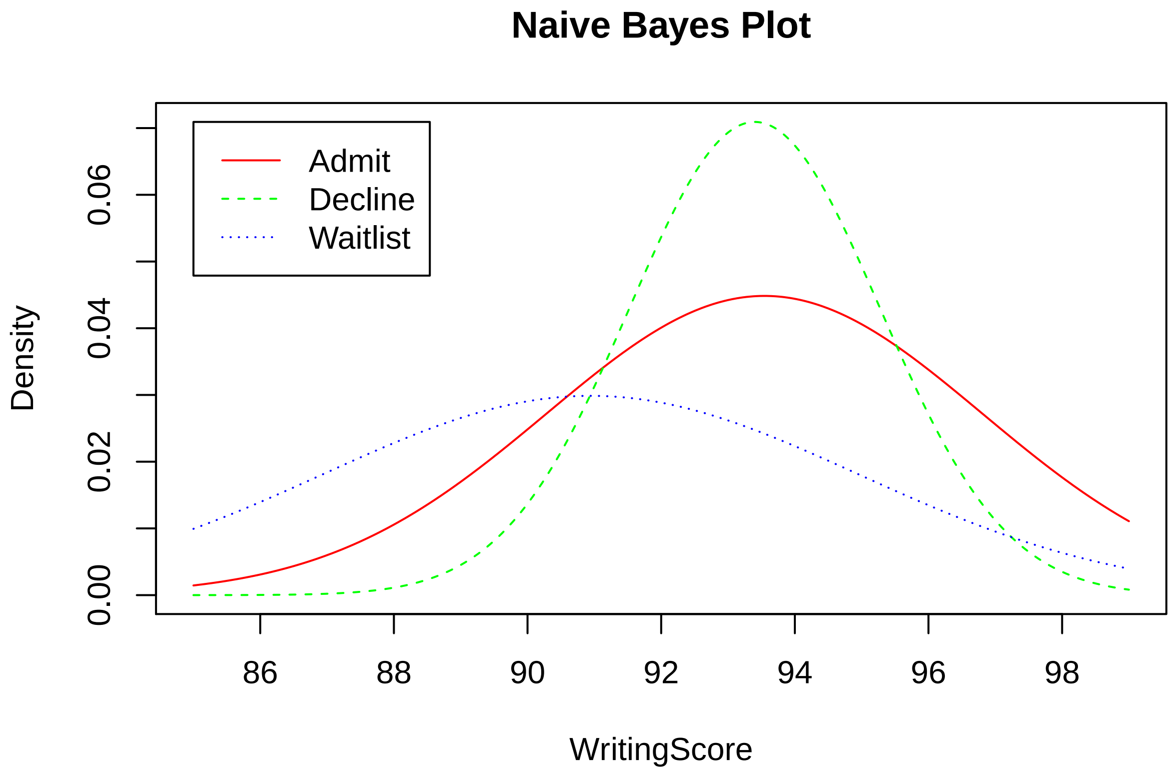

Naive Bayes Plot
Admit
Decline
Waitlist
Female
Male
var
grouping

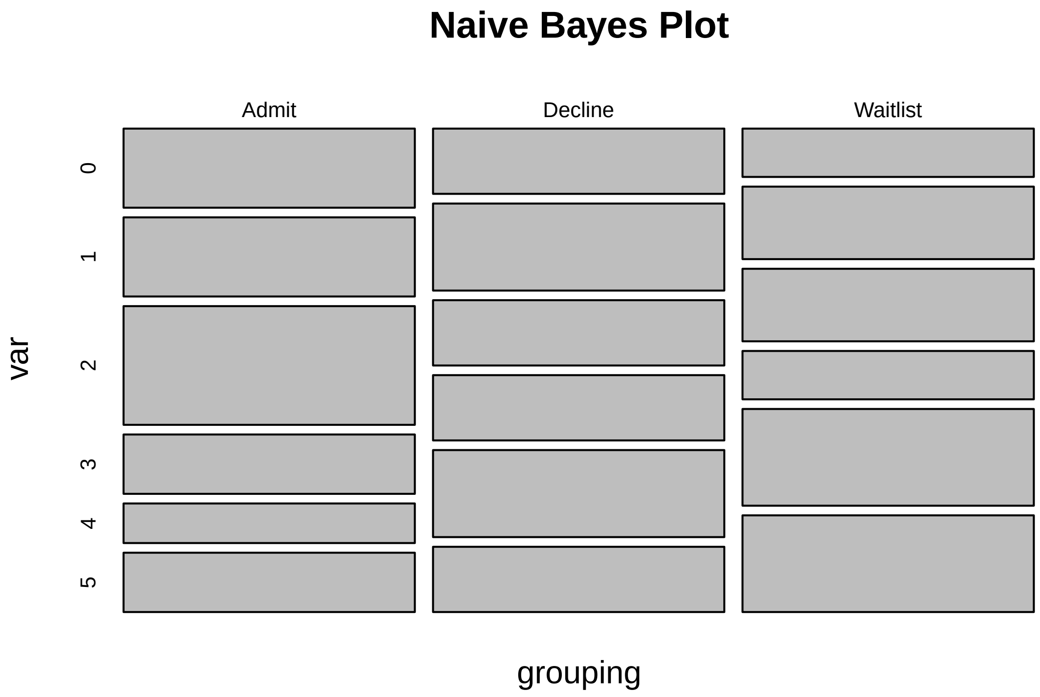

Naive Bayes Plot
Admit
Decline
Waitlist
var
Alabama
California
Colorado
Florida
grouping


Naive Bayes Plot
Admit
Decline
Waitlist
Density
0.0
0.5
1.0
1.5
3.0
3.2
3.4
3.6
3.8
GPA


Naive Bayes Plot
Admit
Decline
Waitlist
Density
0.00
0.05
0.10
0.15
0
1
2
3
4
WorkExp


Naive Bayes Plot
Admit
Decline
Waitlist
Density
0.000
0.010
0.020
0.030
750
800
850
900
950
TestScore


Naive Bayes Plot
Admit
Decline
Waitlist
Density
0.00
0.02
0.04
0.06
86
88
90
92
94
96
98
WritingScore



```
##################################################
##
##          Support Vector Machines
##
########################################################
##
head(Training_Set)  ## Notice that column 1 is the label
```

```
##   Decision Gender       State  GPA WorkExp TestScore WritingScore VolunteerLevel
## 2    Admit Female    Florida 3.55     0.0       962           97              0
## 4    Admit Female   Colorado 3.60     0.9       969           97              2
## 5    Admit Female   Colorado 3.60     1.2       967           94              2
## 6    Admit Female California 3.66     0.9       956           89              1
## 7    Admit Female California 3.70     1.2       969           94              2
## 8    Admit Female California 3.70     2.7       799           97              5
```

```
## Notice that columns 4, 5, 6, and 7 are numeric.
head(Testing_Set_No_Labels)
```

```
##    Gender      State  GPA WorkExp TestScore WritingScore VolunteerLevel
## 10 Female    Florida 3.77     1.4       969           99              4
## 54 Female    Florida 3.56     1.3       869           94              3
## 52 Female    Florida 3.55     2.0       853           94              1
## 53 Female    Georgia 3.56     1.0       866           89              1
## 72   Male   Colorado 3.80     0.8       969           93              1
## 84   Male California 3.42      0.7       869           94              3
```

```r
head(MyTestLabels)
```

```
## [1] Admit    Waitlist Waitlist Waitlist Admit    Waitlist
## Levels: Admit Decline Waitlist
```

```r
str(MyData)
```

```
## 'data.frame':    85 obs. of  9 variables:
##  $ Decision      : Factor w/ 3 levels "Admit","Decline",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender        : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DateSub       : Date, format: "2020-01-11" "2020-01-11" ...
##  $ State         : Factor w/ 11 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
##  $ GPA           : num  3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
##  $ WorkExp       : num  0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
##  $ TestScore     : int  965 962 969 969 967 956 969 799 969 969 ...
##  $ WritingScore  : int  94 97 93 97 94 89 94 97 93 99 ...
##  $ VolunteerLevel: Factor w/ 6 levels "0","1","2","3",..: 2 1 1 3 3 2 3 6 1 5 ...
```

```r
#names(MyData)
## Create NEW DFs with just the numeric data
SVM_Training_Data<-Training_Set[,c(1,4,6)]
SVM_Test_Data_noLabel<-Testing_Set_No_Labels[,c(3,5)]  ## Why 3 and 5??
## Because in this testset, that is where these columns are GPA and TestScore

## Recall that Support Vector Machines ONLY WORK ON NUMERIC data
## Consider our LABEL - this is "Decision"
## This must be type factor - and it is - which is good.
## Next, like all other goals in R - there are always
## many different library options.

library(tidyverse)    # data manipulation and visualization
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ------------------------------------------------------- tidyverse 1.3.0 --
```

```
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

```
## Warning: package 'purrr' was built under R version 3.5.3
```

```
## Warning: package 'stringr' was built under R version 3.5.3
```

```
## Warning: package 'forcats' was built under R version 3.5.3
## -- Conflicts ---------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::arrange()         masks plyr::arrange()
## x lubridate::as.difftime() masks base::as.difftime()
## x stringr::boundary()      masks strucchange::boundary()
## x randomForest::combine()  masks dplyr::combine()
## x purrr::compact()         masks plyr::compact()
## x dplyr::count()           masks plyr::count()
## x lubridate::date()        masks base::date()
## x tidyr::expand()          masks Matrix::expand()
## x dplyr::failwith()        masks plyr::failwith()
## x dplyr::filter()          masks stats::filter()
## x dplyr::id()              masks plyr::id()
## x arules::intersect()      masks lubridate::intersect(), base::intersect()
## x dplyr::lag()             masks stats::lag()
## x purrr::lift()            masks caret::lift()
## x purrr::map()             masks mclust::map()
## x randomForest::margin()   masks ggplot2::margin()
## x dplyr::mutate()          masks ggpubr::mutate(), plyr::mutate()
## x tidyr::pack()            masks Matrix::pack()
## x arules::recode()         masks dplyr::recode()
## x dplyr::rename()          masks plyr::rename()
## x MASS::select()           masks dplyr::select()
## x arules::setdiff()        masks lubridate::setdiff(), base::setdiff()
## x dplyr::summarise()       masks plyr::summarise()
## x dplyr::summarize()       masks plyr::summarize()
## x arules::union()          masks lubridate::union(), base::union()
## x tidyr::unpack()          masks Matrix::unpack()
```

```
library(kernlab)      # SVM methodology
```

```
## Warning: package 'kernlab' was built under R version 3.5.3
```

```
##
## Attaching package: 'kernlab'
```

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

```
## The following object is masked from 'package:modeltools':
##
##     prior
```

```
## The following object is masked from 'package:arules':
##
##     size
```

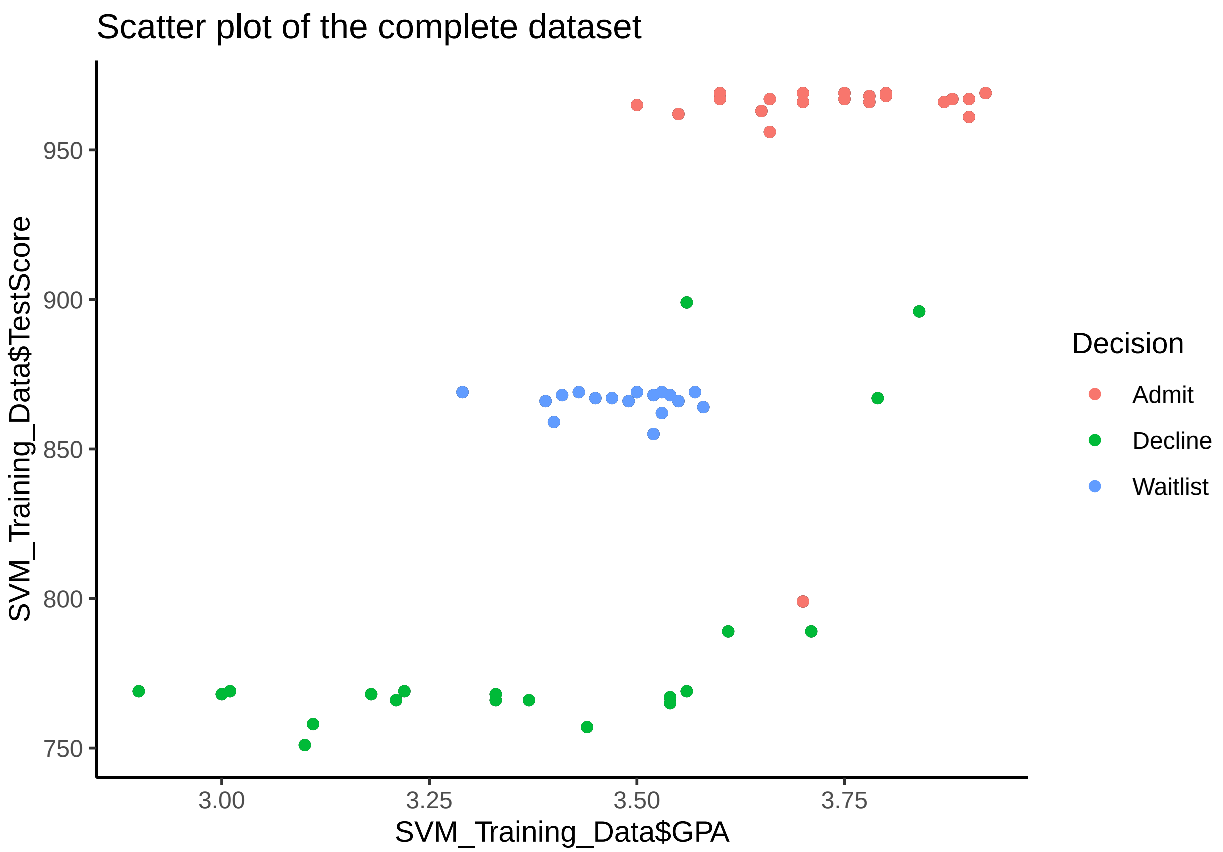
```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```r
library(e1071)        # SVM methodology
#install.packages("ISLR")
library(ISLR)         # contains example data set "Khan"
```

```
## Warning: package 'ISLR' was built under R version 3.5.3
```

```r
library(RColorBrewer) # customized coloring of plots

## Let's start with a fun example
## Let's use only two of our numeric variables and let's
## SEE what the SVM linear predictor looks like

### NOTES --------------------------------------
## The "x" Points are the support vectors
## The "o"points are the other points
## which do not affect the calculation of the linear sep
##----------------------------------------------------

##-------
## plot the dataset
##-----------------------------
# plot the complete dataset
ggplot(data= SVM_Training_Data,
       aes(x=SVM_Training_Data$GPA,
           y= SVM_Training_Data$TestScore)) +
  geom_point() +
  geom_point(aes(color=Decision))+
  scale_shape_manual(values=c(1,3)) +
  ggtitle("Scatter plot of the complete dataset")
```

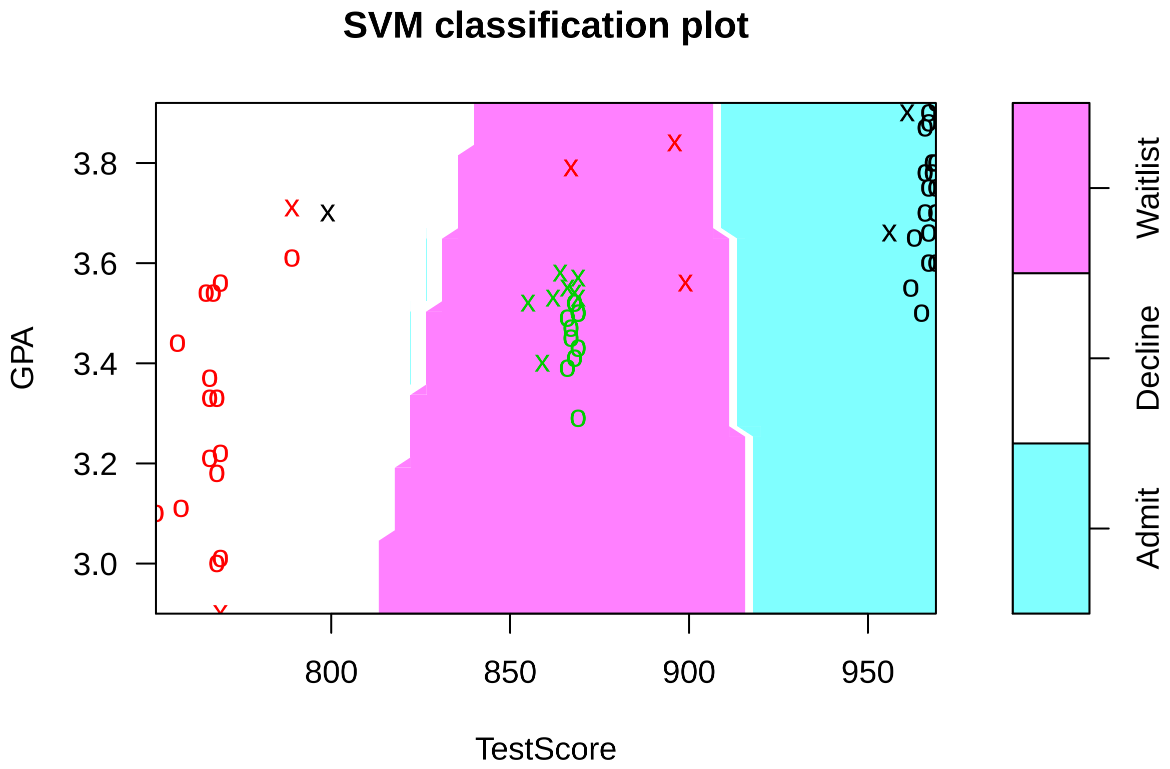
```
## Warning: Use of `SVM_Training_Data$GPA` is discouraged. Use `GPA` instead.

## Warning: Use of `SVM_Training_Data$TestScore` is discouraged. Use `TestScore`
## instead.

## Warning: Use of `SVM_Training_Data$GPA` is discouraged. Use `GPA` instead.

## Warning: Use of `SVM_Training_Data$TestScore` is discouraged. Use `TestScore`
## instead.
```



```r
## from e1071
SVM_fit1 <- svm(Decision~.,  data = SVM_Training_Data, kernel = "linear", scale = FALSE)
SVM_fit1
```

```
##
## Call:
## svm(formula = Decision ~ ., data = SVM_Training_Data, kernel = "linear",
##     scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  16
```

```r
# Plot Results
plot(SVM_fit1, SVM_Training_Data)
```



```
## This worked really well! We can SEE the seperating lines.
## Notice that R used TWO (2) SVMS!! WHY??

### Let's see a confusion matrix

SVM_Pred1 <- predict(SVM_fit1, SVM_Test_Data_noLabel)

## Basic Confusion Matrix
table(SVM_Pred1,MyTestLabels)
```

```
##             MyTestLabels
## SVM_Pred1  Admit Decline Waitlist
##   Admit       11       0        0
##   Decline      0       7        0
##   Waitlist     0       0        7
```

```
##---------------------------------------------
## Pretty confusion matrix:
##-------------------------------------------------------

(MyResults2 <- data.frame(Predicted=SVM_Pred1,Actual=MyTestLabels))
```

```
##    Predicted   Actual
## 10     Admit    Admit
## 54  Waitlist Waitlist
## 52  Waitlist Waitlist
## 53  Waitlist Waitlist
## 72     Admit    Admit
## 84  Waitlist Waitlist
## 1      Admit    Admit
## 20   Decline  Decline
## 82   Decline  Decline
## 41  Waitlist Waitlist
## 86  Waitlist Waitlist
## 42  Waitlist Waitlist
## 22   Decline  Decline
## 69     Admit    Admit
## 75   Decline  Decline
## 61     Admit    Admit
## 21   Decline  Decline
## 80   Decline  Decline
## 13     Admit    Admit
## 16     Admit    Admit
## 73   Decline  Decline
## 71     Admit    Admit
## 11     Admit    Admit
## 3      Admit    Admit
## 14     Admit    Admit
```

```
## Basic Confusion Matrix
(MyTable2<-table(SVM_Pred1,MyTestLabels))
```

```
##           MyTestLabels
## SVM_Pred1  Admit Decline Waitlist
##   Admit       11       0        0
##   Decline      0       7        0
##   Waitlist     0       0        7
```

```
str(MyTable2)
```

```
##  'table' int [1:3, 1:3] 11 0 0 0 7 0 0 0 7
##  - attr(*, "dimnames")=List of 2
##   ..$ SVM_Pred1   : chr [1:3] "Admit" "Decline" "Waitlist"
##   ..$ MyTestLabels: chr [1:3] "Admit" "Decline" "Waitlist"
```

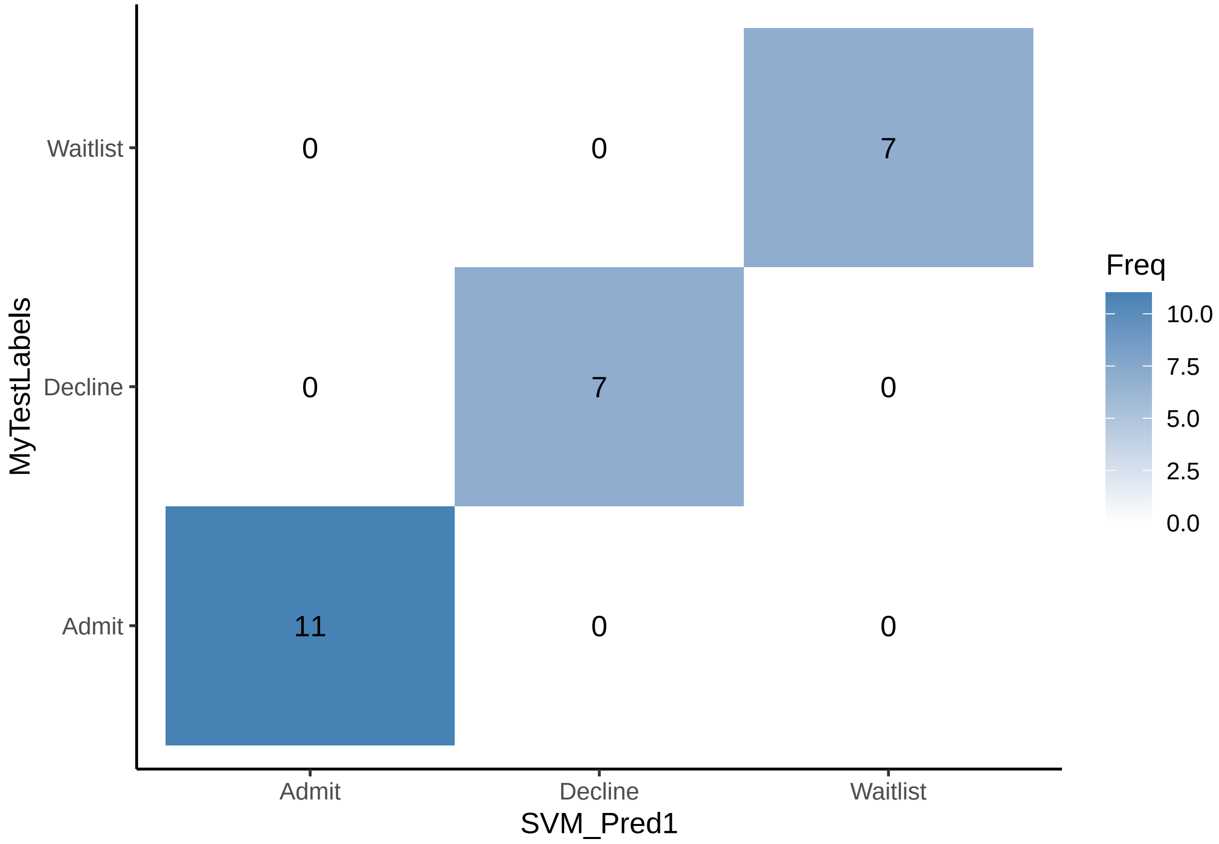
```
## Create a DF from the table to use in the heat map below...
(MyTable_DF<-as.data.frame(MyTable2))
```

```
##   SVM_Pred1 MyTestLabels Freq
## 1     Admit        Admit   11
## 2   Decline        Admit    0
## 3  Waitlist        Admit    0
## 4     Admit      Decline    0
## 5   Decline      Decline    7
## 6  Waitlist      Decline    0
## 7     Admit     Waitlist    0
## 8   Decline     Waitlist    0
## 9  Waitlist     Waitlist    7
```

```r
str(MyTable_DF)
```

```
## 'data.frame':    9 obs. of  3 variables:
##  $ SVM_Pred1   : Factor w/ 3 levels "Admit","Decline",..: 1 2 3 1 2 3 1 2 3
##  $ MyTestLabels: Factor w/ 3 levels "Admit","Decline",..: 1 1 1 2 2 2 3 3 3
##  $ Freq        : int  11 0 0 0 7 0 0 0 7
```

```r
## BE CAREFUL - you need the steps above to REFORMAT
## So that you can create the heat map.

## Using ggplot heatmap to build a confusion matrix
ggplot(MyTable_DF, aes(x=SVM_Pred1, y=MyTestLabels, fill=Freq)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "steelblue")+
  geom_text(aes(label=Freq))
```



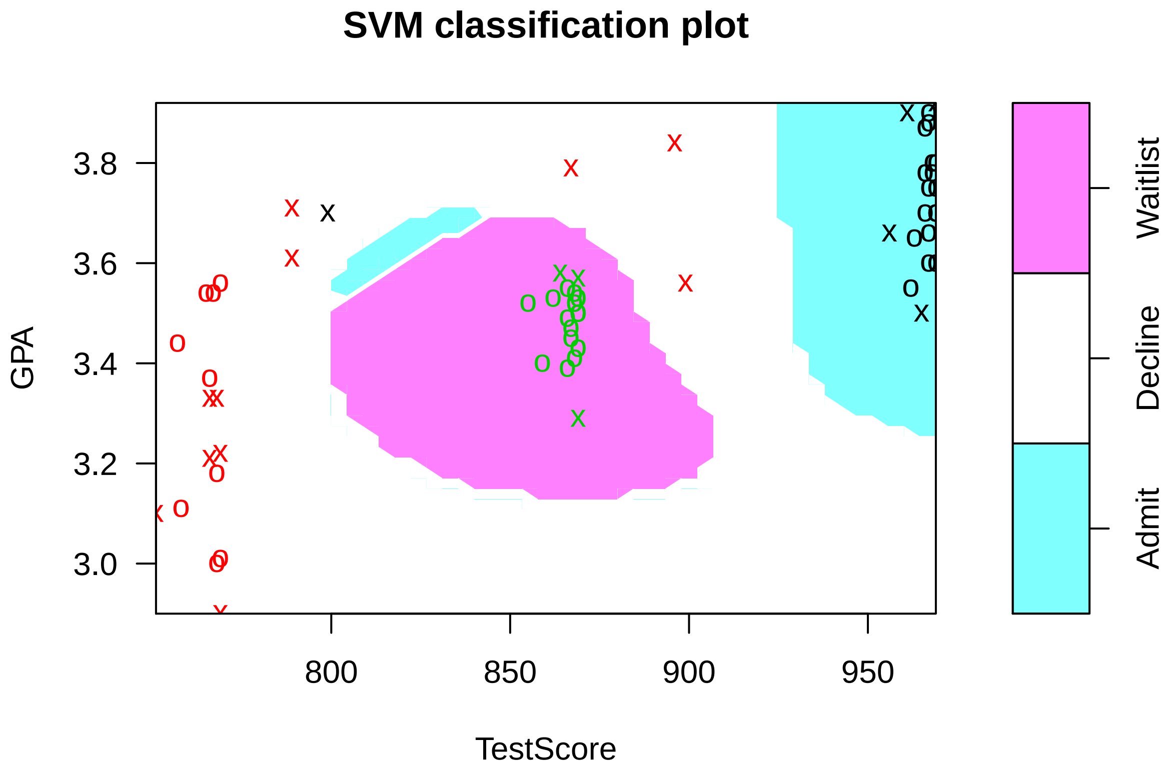
```r
##########################################
## Tuning - finding the best cost C
##########################################
# find optimal cost of misclassification
BestCost <- tune(svm, Decision~., data = SVM_Training_Data, kernel = "linear",
                 ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))

(BEST <- BestCost$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = Decision ~ ., data = SVM_Training_Data,
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)), kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  46
```

```
###################################
## Other kernels....include radial, polynomial, etc.
##########################################################
### IT IS VERY IMPORTANT TO NORMALIZE DATA
## WHEN USING AN SVM!! (WHy??)
SVM_fit2 <- svm(Decision~.,  data = SVM_Training_Data,
                kernel = "radial",
                scale = TRUE,
                cost=10,
                gamma=1)
SVM_fit2
```

```
##
## Call:
## svm(formula = Decision ~ ., data = SVM_Training_Data, kernel = "radial",
##     cost = 10, gamma = 1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  19
```

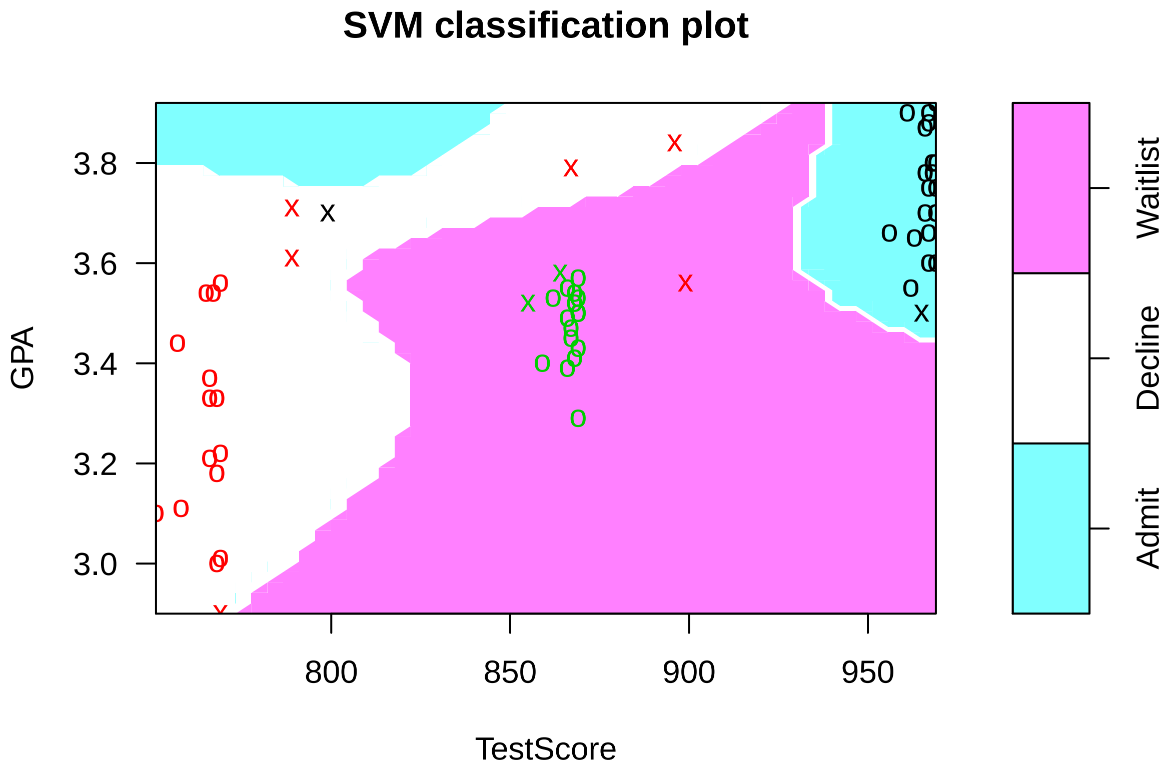
```
plot(SVM_fit2, SVM_Training_Data)
```



```
# ---------------------------------------POLYNOMIAL ---------
SVM_fit3 <- svm(Decision~.,  data = SVM_Training_Data,
                kernel = "polynomial",
                scale = TRUE,
                cost=10,
                gamma=1)
SVM_fit3
```

```
##
## Call:
## svm(formula = Decision ~ ., data = SVM_Training_Data, kernel = "polynomial",
##     cost = 10, gamma = 1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  10
```

```
# Plot Results
plot(SVM_fit3, SVM_Training_Data)
```



```
##################################################################
## A look at other options for prediction and confusion matrices
#################################################################

##################################################################
### Running cross validation, and tuning with cost
##  to create the best model............
##################################################################

BestCost2 <- tune(svm, Decision~., data = SVM_Training_Data, kernel = "linear",
                  ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))

summary(BestCost2)
```

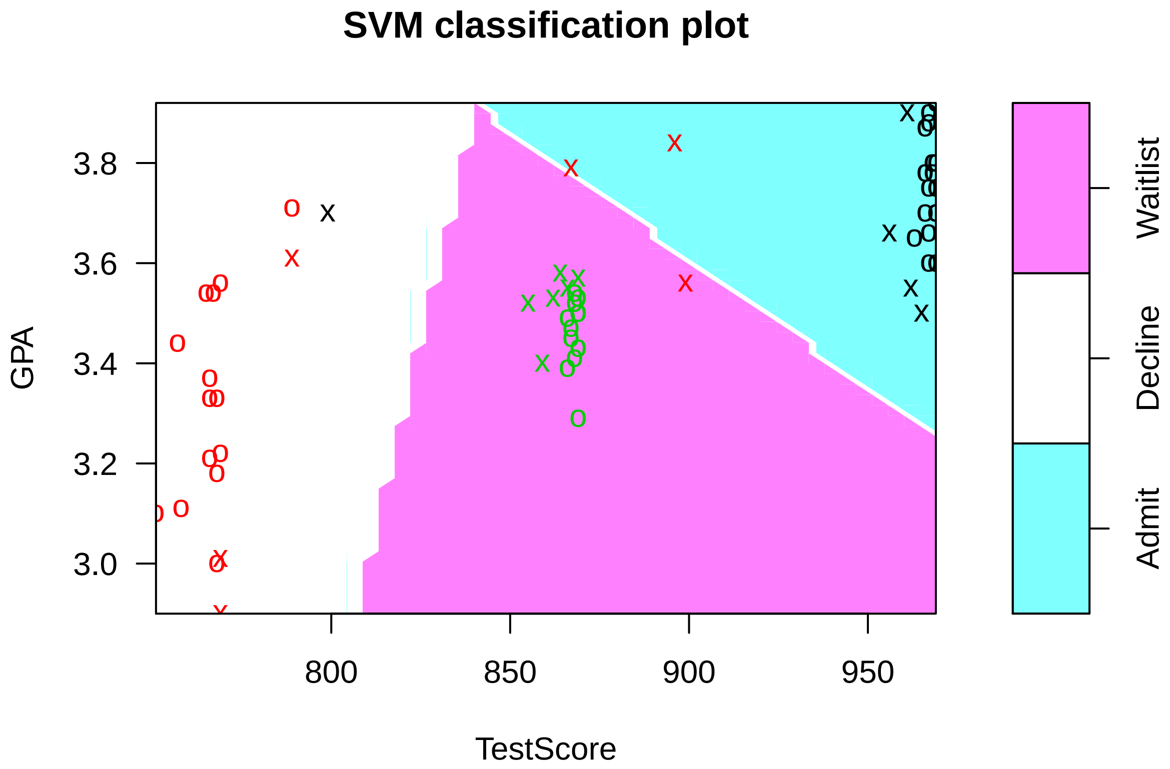
```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##     5
##
## - best performance: 0.05
##
## - Detailed performance results:
```

```
##    cost      error dispersion
## 1 1e-03 0.68333333 0.19953650
## 2 1e-02 0.45000000 0.20861093
## 3 1e-01 0.08333333 0.11785113
## 4 1e+00 0.06666667 0.08606630
## 5 5e+00 0.05000000 0.08050765
## 6 1e+01 0.05000000 0.08050765
## 7 1e+02 0.05000000 0.08050765
```

```
SVM_pred3 <- predict(BestCost2$best.model, SVM_Test_Data_noLabel)
confusionMatrix(SVM_pred3, MyTestLabels)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Admit Decline Waitlist
##   Admit       11       0        0
##   Decline      0       7        0
##   Waitlist     0       0        7
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.8628, 1)
##     No Information Rate : 0.44
##     P-Value [Acc > NIR] : 1.22e-09
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Admit Class: Decline Class: Waitlist
## Sensitivity                  1.00           1.00            1.00
## Specificity                  1.00           1.00            1.00
## Pos Pred Value               1.00           1.00            1.00
## Neg Pred Value               1.00           1.00            1.00
## Prevalence                   0.44           0.28            0.28
## Detection Rate               0.44           0.28            0.28
## Detection Prevalence         0.44           0.28            0.28
## Balanced Accuracy            1.00           1.00            1.00
```

```
plot(BestCost2$best.model, data=SVM_Training_Data)
```



```
##########################################################
##             Using the kernlab library for SVMs
########################################################

#library(kernlab)
SVM_fit4 <- ksvm(Decision ~ ., data = SVM_Training_Data,
                 kernel = "vanilladot", scaled=TRUE)
```

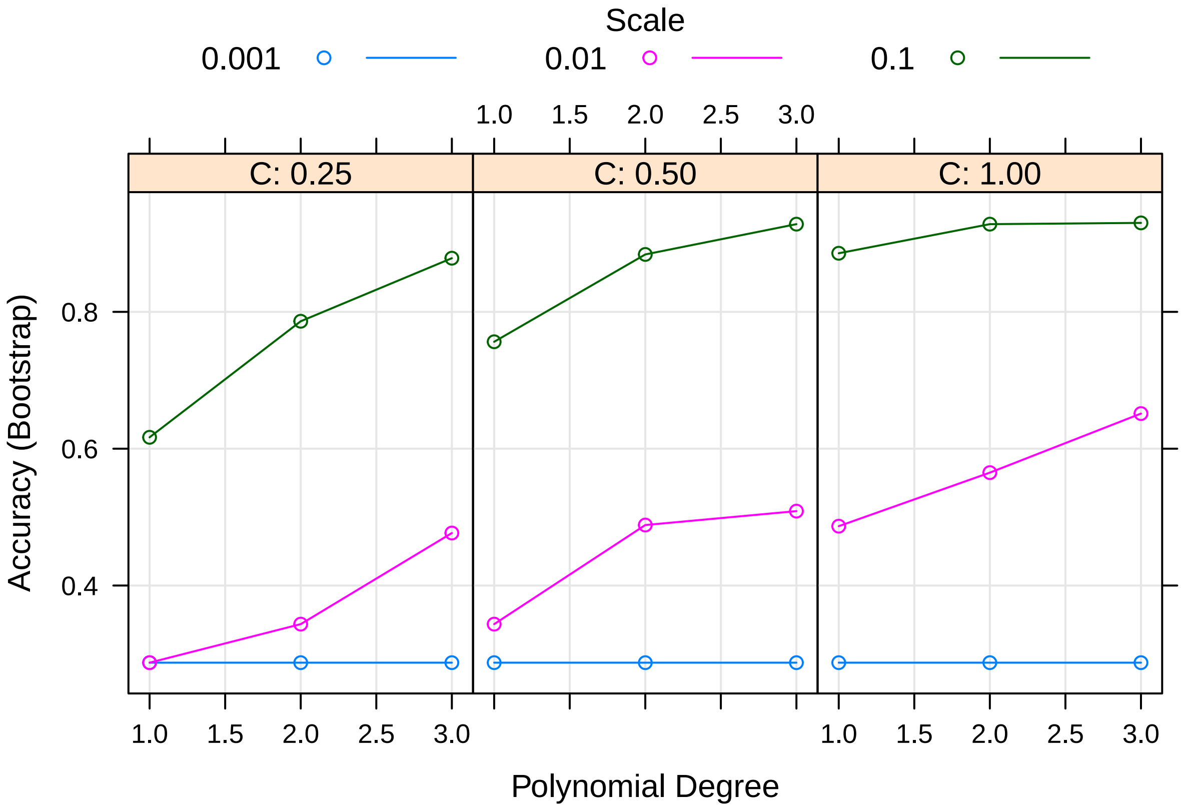
```
##  Setting default kernel parameters
```

```
#vanilladot is the Euclidean inner product kernel
# option  kernel="rbfdot"


## The plot function for ksvm only works for 2-class problems
## Supports only BINARY classification.
## https://rdrr.io/cran/kernlab/man/ksvm.html
## plot(SVM_fit4, data=SVM_Training_Data)

####################################################
## Using *train* to look at parameters
####################################################
(SVM5 <- train(Decision ~., SVM_Training_Data,
                 method="svmPoly"))
```

```
## Support Vector Machines with Polynomial Kernel
```

```
## 
## 60 samples
##  2 predictor
##  3 classes: 'Admit', 'Decline', 'Waitlist' 
## 
## No pre-processing
## Resampling: Bootstrapped (25 reps) 
## Summary of sample sizes: 60, 60, 60, 60, 60, 60, ... 
## Resampling results across tuning parameters:
## 
##   degree  scale  C     Accuracy   Kappa      
##   1       0.001  0.25  0.2872676  0.006423358
##   1       0.001  0.50  0.2872676  0.006423358
##   1       0.001  1.00  0.2872676  0.006423358
##   1       0.010  0.25  0.2872676  0.006423358
##   1       0.010  0.50  0.3435835  0.076329934
##   1       0.010  1.00  0.4867606  0.258531224
##   1       0.100  0.25  0.6166747  0.439650184
##   1       0.100  0.50  0.7562925  0.642920754
##   1       0.100  1.00  0.8856508  0.833263779
##   2       0.001  0.25  0.2872676  0.006423358
##   2       0.001  0.50  0.2872676  0.006423358
##   2       0.001  1.00  0.2872676  0.006423358
##   2       0.010  0.25  0.3435835  0.076329934
##   2       0.010  0.50  0.4884273  0.261031224
##   2       0.010  1.00  0.5650143  0.366653412
##   2       0.100  0.25  0.7862727  0.687706031
##   2       0.100  0.50  0.8839117  0.829374903
##   2       0.100  1.00  0.9281995  0.889797468
##   3       0.001  0.25  0.2872676  0.006423358
##   3       0.001  0.50  0.2872676  0.006423358
##   3       0.001  1.00  0.2872676  0.006423358
##   3       0.010  0.25  0.4767451  0.250628709
##   3       0.010  0.50  0.5087367  0.289487309
##   3       0.010  1.00  0.6513254  0.490050428
##   3       0.100  0.25  0.8784335  0.821190705
##   3       0.100  0.50  0.9281995  0.890158501
##   3       0.100  1.00  0.9300177  0.892588890
## 
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were degree = 3, scale = 0.1 and C = 1.
```

```
## Visualize Accuracy
plot(SVM5)
```

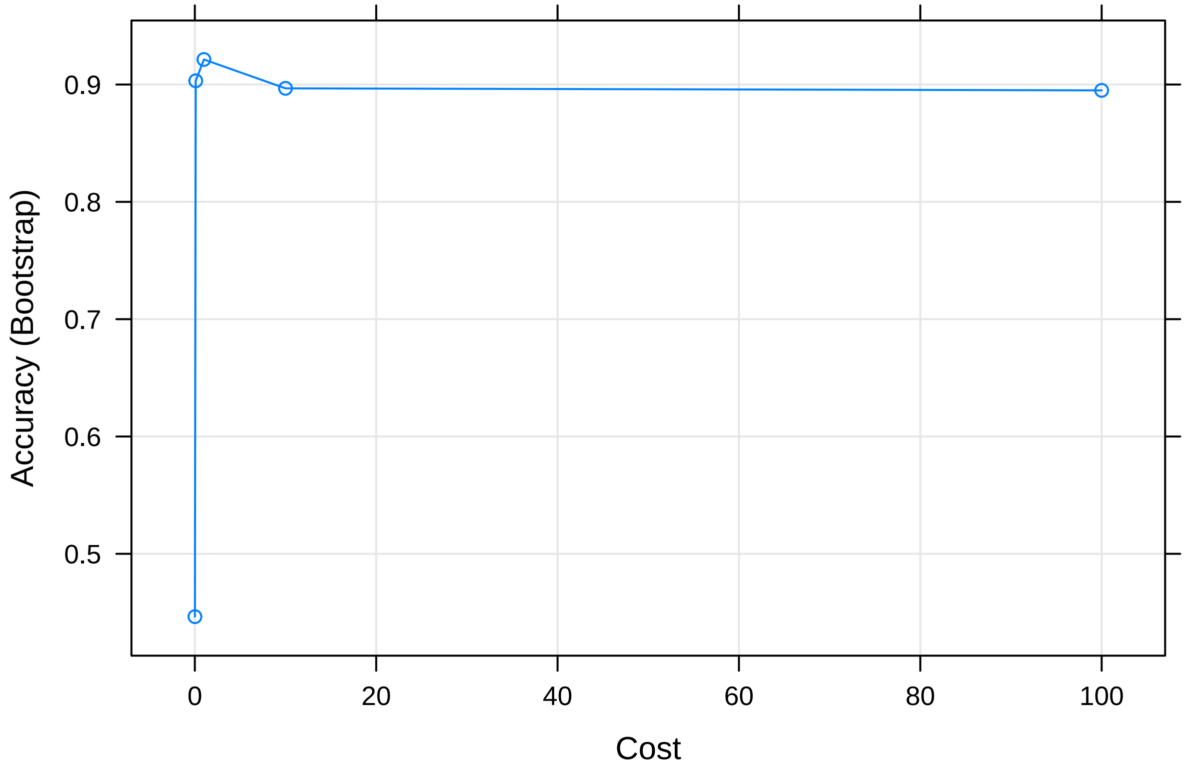


```
###################################################
##
##    Cross validation
##
######################################################
CrossVal<-trainControl(method="repeatedcv", repeats=5, classProbs = TRUE)
## Include trying different costs
MyCostGrid<-expand.grid(C=c(.01, .1, 1, 10, 100))

(MySVM6<-train(Decision~.,
               data = SVM_Training_Data,
               method="svmLinear",
                   preProc=c("center","scale"),
                   tuneGrid=MyCostGrid,
                   metric="Accuracy"))
```

```
## Support Vector Machines with Linear Kernel
##
## 60 samples
##  2 predictor
##  3 classes: 'Admit', 'Decline', 'Waitlist'
##
## Pre-processing: centered (2), scaled (2)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 60, 60, 60, 60, 60, 60, ...
## Resampling results across tuning parameters:
```

```
##
##   C      Accuracy   Kappa
##   1e-02  0.4464228  0.2553515
##   1e-01  0.9031624  0.8505015
##   1e+00  0.9213443  0.8773272
##   1e+01  0.8967546  0.8398656
##   1e+02  0.8950264  0.8370196
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 1.
```

```
plot(MySVM6)
```



```
#################################################################
## SVM Tutorials
#################################################################
## references for SVMs and math
## https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html
## http://datascienceandme.com/topics/RSupportVectorMachine.html
## https://medium.com/nyu-a3sr-data-science-team/support-vector-machines-and-wine-cef59ad38b41
## https://cran.r-project.org/web/packages/kernlab/kernlab.pdf
## https://www.isical.ac.in/~arnabc/multi/lecsvm4.html
## https://rpubs.com/ryankelly/svm
## #################################################################
```

```r
################################################################
##
## Regression - Linear and non-linear
##
#################################################################

## Linear regression is used to PREDICT a value
## of a variable (called dependent or outcome or response)
## The goal is to create a linear equation
## that best estimates the values of the
## response.

## Example: Y = b +b1X + error
## Multiple Linear Regression:  Y=b+b1X1+b2x2+...
## The "b" values are the coefficients

head(MyData)
```

```
##   Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1    Admit Female 2020-01-11    Florida 3.54     0.7       965           94
## 2    Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3    Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4    Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5    Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6    Admit Female 2019-11-03 California 3.66     0.9       956           89
##   VolunteerLevel
## 1              1
## 2              0
## 3              0
## 4              2
## 5              2
## 6              1
```

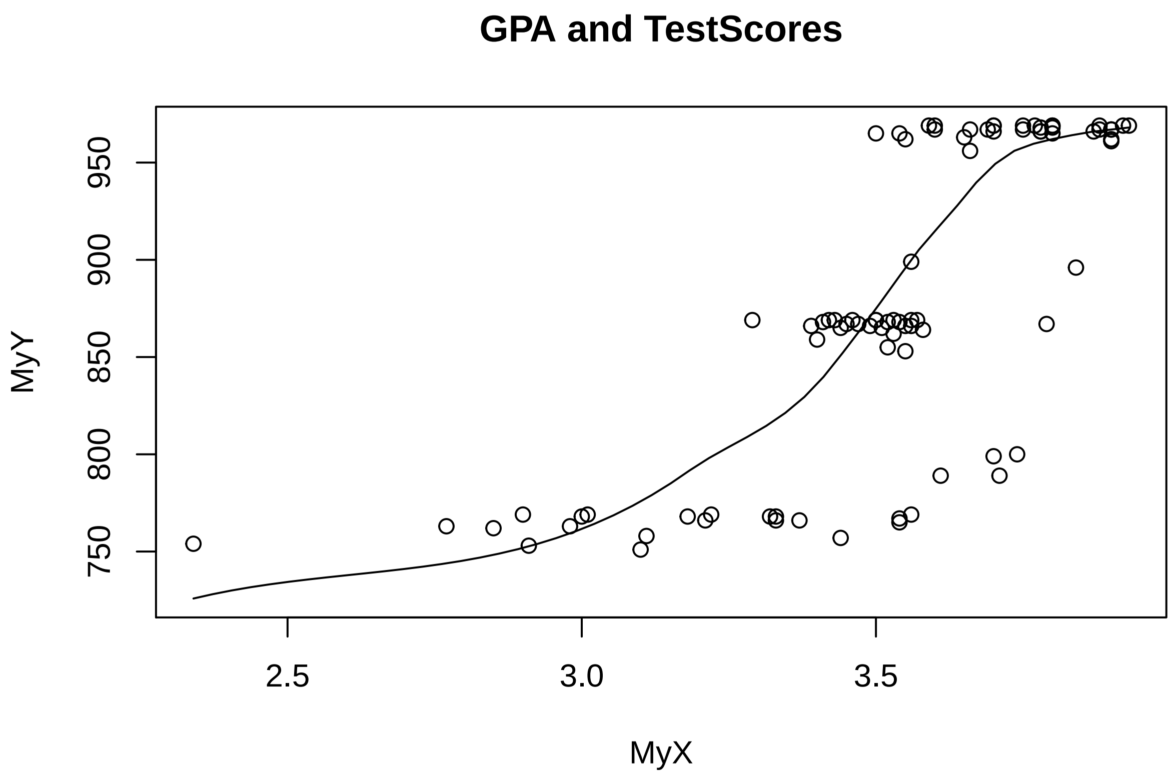
```r
## Recall that regression is math
## Therefore, it only works on numeric data.
(MyX<-MyData[,5]) ## Let's make this 2D for now
```

```
##  [1] 3.54 3.55 3.59 3.60 3.60 3.66 3.70 3.70 3.75 3.77 3.78 3.78 3.80 3.88 3.90
## [16] 3.90 3.90 3.75 2.34 2.85 2.98 3.01 3.18 3.21 3.33 3.33 3.37 3.44 3.54 3.54
## [31] 3.56 3.61 3.71 3.79 3.84 3.39 3.40 3.41 3.43 3.44 3.46 3.47 3.49 3.50 3.50
## [46] 3.52 3.53 3.53 3.54 3.55 3.55 3.56 3.56 3.57 3.58 3.50 3.65 3.66 3.69 3.70
## [61] 3.70 3.78 3.80 3.80 3.87 3.88 3.90 3.92 3.93 3.80 2.77 2.90 2.91 3.00 3.10
## [76] 3.11 3.22 3.32 3.56 3.74 3.29 3.42 3.45 3.51 3.52
```

```r
(MyY<-MyData[,7])
```

```
##  [1] 965 962 969 969 967 956 969 799 969 969 966 968 965 969 961 967 967 967 754
## [20] 762 763 769 768 766 766 768 766 757 765 767 769 789 789 867 896 866 859 868
## [39] 869 865 869 867 866 869 869 868 869 862 868 866 853 866 869 869 864 965 963
## [58] 967 967 966 969 966 969 968 966 967 962 969 969 969 763 769 753 768 751 758
## [77] 769 768 899 800 869 869 867 865 855
```

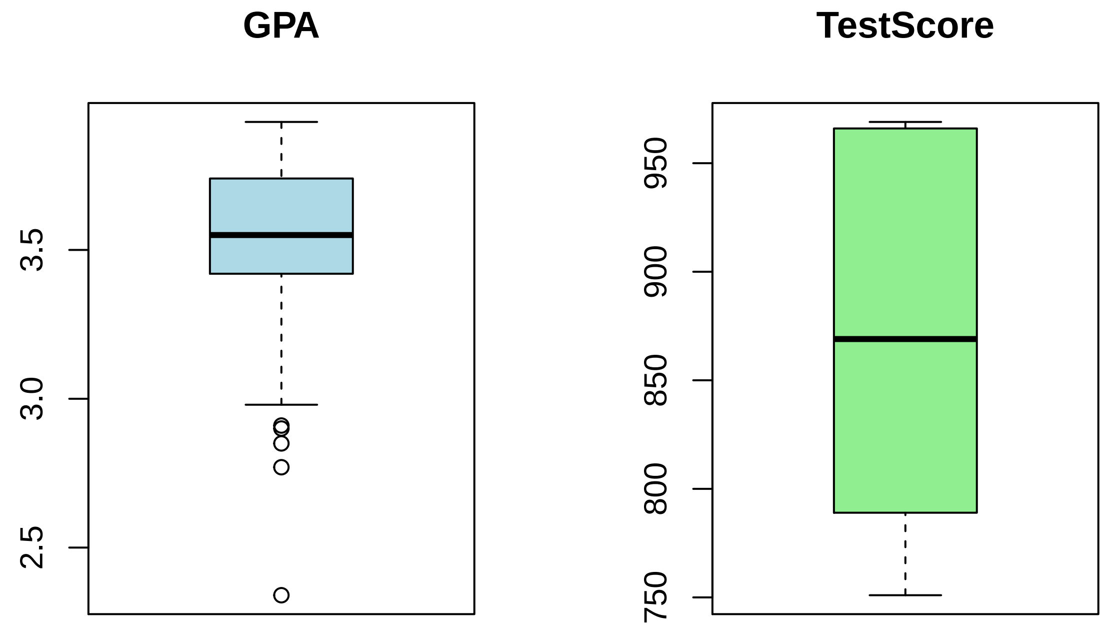
```
scatter.smooth(x=MyX, y=MyY, main="GPA and TestScores")   # scatterplot
```



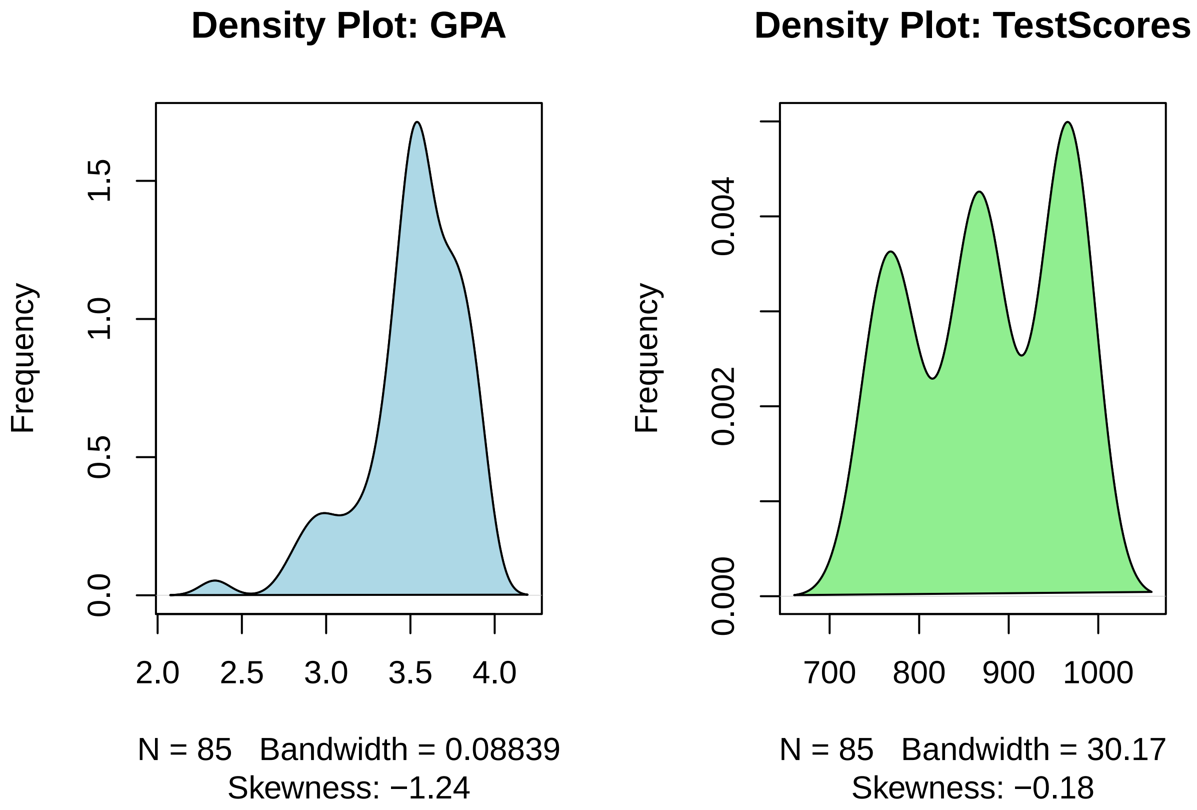
```
## Checking for outliers and looking at the data
par(mfrow=c(1, 2))  # subplot graph area in 2 columns
boxplot(MyX, main="GPA", col="lightblue")

## These are the points that are "far" from the median.
## However, they are not outliers!
(Stats<-boxplot.stats(MyX)$out)
```

```
## [1] 2.34 2.85 2.77 2.90 2.91
```

```
## a vector of length 5, containing the extreme of the lower whisker
#https://www.rdocumentation.org/packages/grDevices/versions/3.6.2/topics/boxplot.stats

boxplot(MyY, main="TestScore", col="lightgreen")
```



```
########-------------- DENSITY ------------
#library(e1071)
par(mfrow=c(1, 2))  # 1 row and 2 columns
plot(density(MyX), main="Density Plot: GPA",
     ylab="Frequency",
     sub=paste("Skewness:", round(e1071::skewness(MyX), 2)))
# density plot for GPA
polygon(density(MyX), col="lightblue")

plot(density(MyY), main="Density Plot: TestScores",
     ylab="Frequency",
     sub=paste("Skewness:", round(e1071::skewness(MyY), 2)))
# density plot for testscores
polygon(density(MyY), col="lightgreen")
```



```
############----------------correlation------------
cor(MyX, MyY)   ## Strong Positive!
```

```
## [1] 0.7530573
```

```
##---------------------------------------------------
##############-----------Build the LINEAR MODEL------
##-------------------------------------------------------

MyLinearModel1 <- lm(MyX ~ MyY)
# build linear regression model on full data
## Can also do this:  MyLinearModel1 <- lm(Var1 ~ Var2, data = yourdata)
print(MyLinearModel1)
```

```
##
## Call:
## lm(formula = MyX ~ MyY)
##
## Coefficients:
## (Intercept)          MyY
##     1.11925      0.00274
```

```
## What does this mean??
## It means this:
```

```
##  Y = 1.11925 + .00274X, where Y is TestScore and X is GPA.
summary(MyLinearModel1)
```

```
##
## Call:
## lm(formula = MyX ~ MyY)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84549 -0.07919 -0.00064  0.10355  0.42859
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.1192454  0.2311760   4.842 5.89e-06 ***
## MyY         0.0027404  0.0002628  10.427  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1964 on 83 degrees of freedom
## Multiple R-squared:  0.5671, Adjusted R-squared:  0.5619
## F-statistic: 108.7 on 1 and 83 DF,  p-value: < 2.2e-16
```

```
## Pr(>|t|) is the sig of the t-test and it is VERY sig in this case
## YES! We CAN predict TestScore with GPA.
## Read more:
## http://r-statistics.co/Linear-Regression.html

############-----------------------------------
##
##   Creating PREDICTIVE linear models
##
###########-------------------------------------------
## Here - we need testing and training data.
##
(MySample<-sample(1:nrow(MyData), 0.8*nrow(MyData)))
```

```
##  [1] 14 64 25 29 39 20 47 66 31 54 32 77 83 68 42 85 58 63 82 15  2 17 55 72 21
## [26] 74 40  9 26 44 57 56 10 69 76 22  5 80 41 60  6 59  3 71  7 73  1 24 52 13
## [51] 30 45 46 53 67  4 81 27 11 49 79 23 43 50 61 37 48 35
```

```
RegrTEST<-MyData[-MySample,c(5,7)]
head(RegrTEST)
```

```
##     GPA TestScore
## 8  3.70       799
## 12 3.78       968
## 16 3.90       967
## 18 3.75       967
## 20 2.34       754
## 29 3.44       757
## This gives 20% test data for GPA and TestScore
RegrTRAIN<-MyData[MySample,c(5,7)]
head(RegrTRAIN)
```

```
##     GPA TestScore
## 14 3.88       969
## 66 3.80       968
## 26 3.33       766
## 30 3.54       765
## 40 3.43       869
## 21 2.85       762
```

```
# Build MODEL on training data ...
(Linear_Pred_Model <- lm(GPA ~ TestScore, data=RegrTRAIN) )
```

```
##
## Call:
## lm(formula = GPA ~ TestScore, data = RegrTRAIN)
##
## Coefficients:
## (Intercept)    TestScore
##    1.158239     0.002684
```

```
## This is the MODEL that was created:
## GPA = 1.288393 + .002558*TestScore

(Linear_Pred <- predict(Linear_Pred_Model, RegrTEST)  )
```

```
##        8       12       16       18       20       29       34       35
## 3.302694 3.756277 3.753593 3.753593 3.181917 3.189969 3.275855 3.485201
##       37       39       52       64       67       72       77       80
## 3.482517 3.487884 3.447626 3.750909 3.750909 3.758961 3.173865 3.219492
##       86
## 3.479833
```

```
## These are the predictions for GPA per the TestScore for each row.

## Build a data frame to compare the true values
## to the predicted values....
True_ValuesDF <- data.frame(cbind(actual=RegrTEST$GPA,
                                  predicted=Linear_Pred))

correlation_accuracy <- cor(True_ValuesDF)
head(True_ValuesDF)
```

```
##    actual predicted
## 8    3.70  3.302694
## 12   3.78  3.756277
## 16   3.90  3.753593
## 18   3.75  3.753593
## 20   2.34  3.181917
## 29   3.44  3.189969
```

```r
## Min-Max Accuracy of the prediction
(min_max_accuracy <-
  mean(apply(True_ValuesDF, 1, min) / apply(True_ValuesDF, 1, max))  )
```

```
## [1] 0.9484696
```

```r
## Impressive!! 96.5% accuracy.

################-------------MultiLinear Regression        ---
###############----------------------------------------------

# Multiple Linear Regression Example
#head(MyData)
ML_Reg <- lm(GPA ~ TestScore + WorkExp + TestScore, data=MyData)
summary(ML_Reg)
```

```
##
## Call:
## lm(formula = GPA ~ TestScore + WorkExp + TestScore, data = MyData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84909 -0.08243  0.00386  0.10224  0.42634
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.1249546  0.2339812   4.808 6.82e-06 ***
## TestScore    0.0027408  0.0002643  10.369  < 2e-16 ***
## WorkExp     -0.0030174  0.0138279  -0.218    0.828
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1975 on 82 degrees of freedom
## Multiple R-squared:  0.5673, Adjusted R-squared:  0.5568
## F-statistic: 53.76 on 2 and 82 DF,  p-value: 1.207e-15
```

```r
## What is the result??
## It is this:
## GPA = 1.125 + .0027*TestScore  - .0030*WorkExp
## P value is nearly 0 at .00000682 - very sig!

# Other Options.......
coefficients(ML_Reg) # model coefficients - we see these above.
```

```
##  (Intercept)    TestScore      WorkExp
##  1.124954635  0.002740786 -0.003017425
```

```r
confint(ML_Reg, level=0.95) # Confidence Intervals for model parameters
```

```
##                    2.5 %      97.5 %
## (Intercept)  0.659491432 1.590417837
## TestScore    0.002214937 0.003266635
## WorkExp     -0.030525460 0.024490610
```

```
fitted(ML_Reg) # predicted values for each row GPA
```

```
##        1        2        3        4        5        6        7        8
## 3.767701 3.761590 3.775646 3.778060 3.771673 3.742430 3.777155 3.306695
##        9       10       11       12       13       14       15       16
## 3.777457 3.776552 3.746302 3.774414 3.764080 3.777759 3.744668 3.763828
##       17       18       20       21       22       23       24       25
## 3.775294 3.766846 3.189093 3.199553 3.214062 3.228394 3.225654 3.219267
##       26       27       28       29       30       31       32       33
## 3.219569 3.227464 3.221681 3.190074 3.218940 3.223818 3.227489 3.283512
##       34       35       36       37       38       39       40       41
## 3.285322 3.496991 3.572552 3.493044 3.473556 3.500336 3.502171 3.474009
##       42       43       44       45       46       47       48       49
## 3.500964 3.494577 3.494552 3.501568 3.496136 3.501844 3.501568 3.477554
##       50       51       52       53       54       55       56       58
## 3.500336 3.491837 3.456810 3.495458 3.502775 3.502473 3.490278 3.767701
##       59       60       61       62       63       64       65       66
## 3.759202 3.768656 3.765639 3.768329 3.769611 3.768933 3.776552 3.772906
##       67       68       69       70       71       72       73       74
## 3.767424 3.770768 3.741374 3.777155 3.778362 3.778362 3.205010 3.229903
##       75       76       77       78       79       80       81       82
## 3.170058 3.226257 3.177552 3.197341 3.222963 3.224748 3.577756 3.313661
##       83       84       85       86       87
## 3.503076 3.504585 3.487034 3.485475 3.460179
```

```
residuals(ML_Reg) # residuals
```

```
##            1            2            3            4            5            6
## -0.227700587 -0.211590428 -0.185646305 -0.178060245 -0.171673446 -0.082430031
##            7            8            9           10           11           12
## -0.077155017  0.393304681 -0.027456760 -0.006551532  0.033698029  0.005585768
##           13           14           15           16           17           18
##  0.035920323  0.102241498  0.155332256  0.136171860  0.124705644 -0.016845566
##           20           21           22           23           24           25
## -0.849093073 -0.349553142 -0.234061887 -0.218394403 -0.045653617 -0.009266818
##           26           27           28           29           30           31
##  0.110431439  0.102535928  0.148319242  0.249926390  0.321060027  0.316181941
##           32           33           34           35           36           37
##  0.332510825  0.326488142  0.424677687  0.293008604  0.267448473 -0.103043640
##           38           39           40           41           42           43
## -0.073556398 -0.090335667 -0.072171225 -0.034008759 -0.040964255 -0.024577456
##           44           45           46           47           48           49
## -0.004552353 -0.001567740  0.003863625  0.018155620  0.028432260  0.052445640
##           50           51           52           53           54           55
##  0.039664333  0.058163330  0.093190058  0.064542419  0.057225290  0.067527032
##           56           58           59           60           61           62
##  0.089722248 -0.267700587 -0.109201591 -0.108656021 -0.075638596 -0.068329175
##           63           64           65           66           67           68
## -0.069611454  0.011067340  0.023448468  0.027094481  0.102576052  0.109231782
##           69           70           71           72           73           74
##  0.158626321  0.142844983  0.151638013  0.021638013 -0.435009611 -0.329903115
##           75           76           77           78           79           80
## -0.260058191 -0.226257102 -0.077551548 -0.087340533 -0.002963037  0.095251610
##           81           82           83           84           85           86
## -0.017756459  0.426339500 -0.213076453 -0.084585165 -0.037033893  0.024525026
##           87
##  0.059820684
```

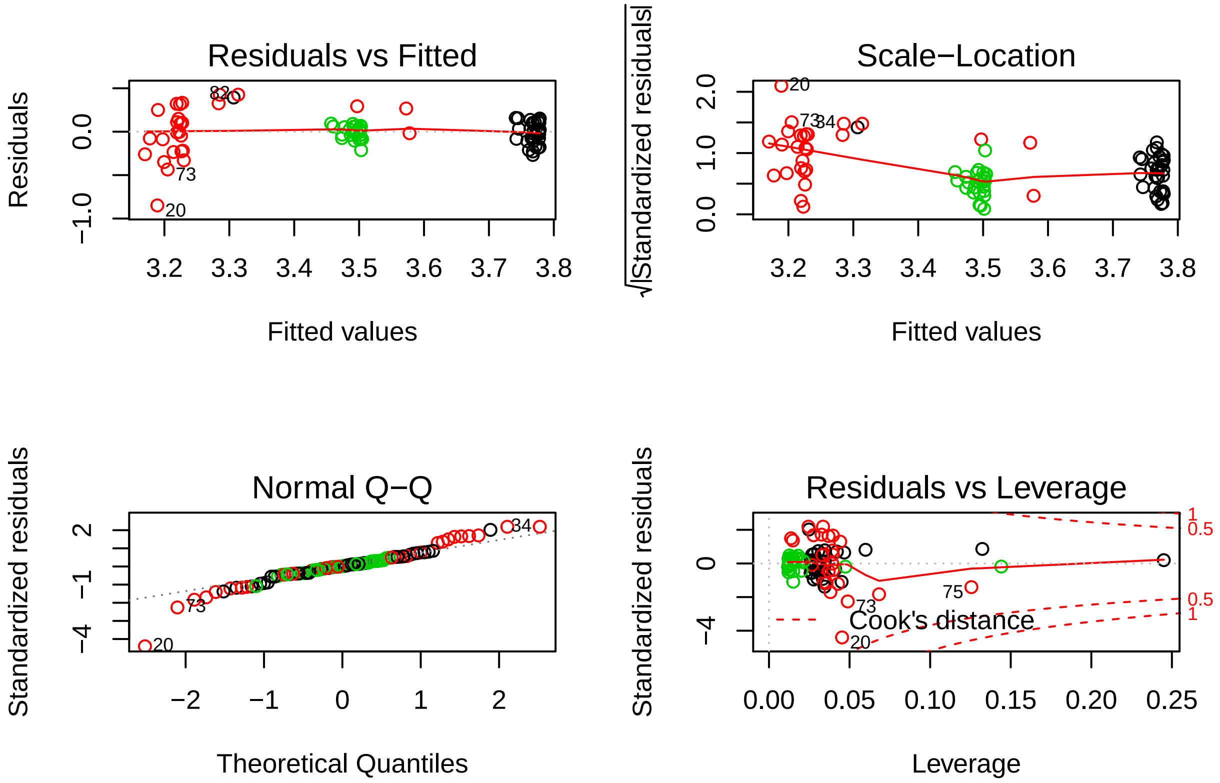
```
anova(ML_Reg) # anova table - the F-test in this case is significant!
```

```
## Analysis of Variance Table
##
## Response: GPA
##           Df Sum Sq Mean Sq  F value Pr(>F)
## TestScore  1 4.1925  4.1925 107.4805 <2e-16 ***
## WorkExp    1 0.0019  0.0019   0.0476 0.8278
## Residuals 82 3.1986  0.0390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vcov(ML_Reg) # covariance matrix for model parameters
```

```
##               (Intercept)     TestScore       WorkExp
## (Intercept)  5.474722e-02 -6.115046e-05 -3.617848e-04
## TestScore   -6.115046e-05  6.987367e-08 -2.561354e-08
## WorkExp     -3.617848e-04 -2.561354e-08  1.912099e-04
```

```
# Plot the Model Diagnostics
layout(matrix(c(1,2,3,4),2,2)) # 4 graphs and 2X 2 graph grid
plot(ML_Reg, col=MyData$Decision)
```



```
## Read More:
## https://www.statmethods.net/stats/regression.html

##################################################################
##
## Neural Networks and Deep Learning
##
##################################################################

## NNs learn by example like all other supervised learning ML
## methods. ANN stands for artificial neural network.
## NNs try to mimic human brain neurons.
## NNs are non-linear, parallel, adaptive (learning), and complex
## NN "adapt" by changing the weights of internal nodes to better
## meet the needs of the problem.
## Interestingly - NNs are used to sole problems that are normally
## easier for humans  - but harder for machines....like knowing
## if someone is sad or finding a sunset in a picture.

## HOW DOES IT WORK
##
## A NN can have a vector of input (many inputs).
## It can weight each input in the vector and can update the
## weights. Each input vector (or data row) is LABELED and so have
## a desired output goal. Like all supervised learning methods, we
## use LABELED TRAINING data to *train* the NN. Then, we use a NON-LABELED
## TEST set to see how well our NN can determine the correct label.
## This idea is the same for all supervised learning methods.
##
## THE FUNCTION OF NN
##
## Y = sum(weight(i)*input(i))+bias
##
## Suppose you have a numeric dataset with variables AGE, WEIGHT, HEIGHT
## Suppose a row (vector) in the dataset is [29, 120, 60].
## Supose we know the LABEL and it is "Female".
## Suppose a different row in the dataset is [34, 210, 75] and
## the label is "Male".
##
## Then, Y is the label - either Male or Female  - but because Y
## will be a number, we will use a THRESHOLD to determine which numbers
## are classified as "Male" and which as "Female".
##
## A possible example (these are made up numbers) might be:
## Y  = (29*.10 + 120*.43 + 60*.67)  = 94.7     <120 so Female
## Y  = (34*.10 + 210*.43 + 75*.67)  = 143.95   >120 so Male
##
## What?? WHy 120??
## Answer:
## This is just an example. The 120 is a threshold. Normally, a sigmoid
## is used to create a threshold.
##

## FIRST (and ignore the warnings)
#install.packages("caret")
#install.packages("nnet")
library(caret)
library(nnet)

## Let's use the SVM datasets because NN also need to be numeric.

(MyTestLabels)
```

```
##  [1] Admit    Waitlist Waitlist Waitlist Admit    Waitlist Admit    Decline
##  [9] Decline  Waitlist Waitlist Waitlist Decline  Admit    Decline  Admit
## [17] Decline  Decline  Admit    Admit    Decline  Admit    Admit    Admit
## [25] Admit
## Levels: Admit Decline Waitlist
```

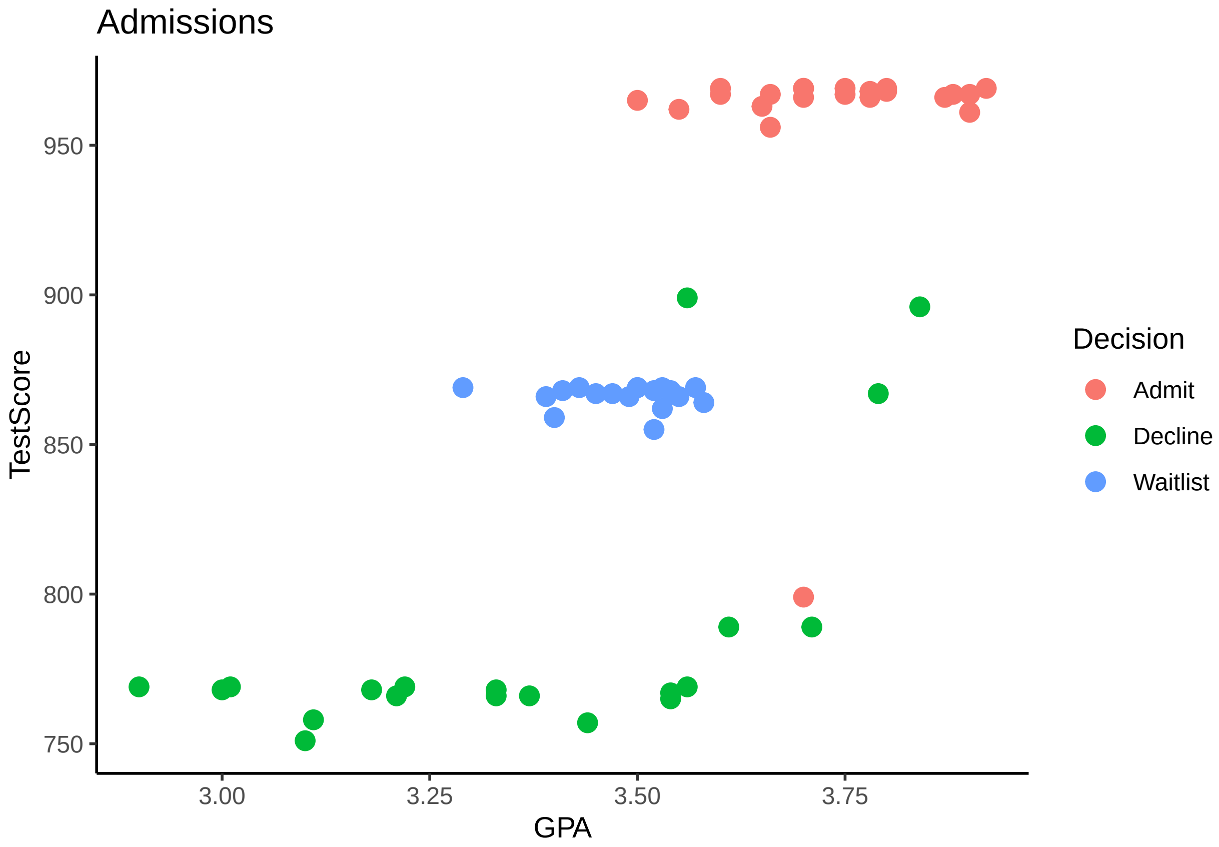
```
head(SVM_Training_Data)
```

```
##   Decision  GPA TestScore
## 2    Admit 3.55       962
## 4    Admit 3.60       969
## 5    Admit 3.60       967
## 6    Admit 3.66       956
## 7    Admit 3.70       969
## 8    Admit 3.70       799
```

```
(SVM_Test_Data_noLabel)
```

```
##     GPA TestScore
## 10 3.77       969
## 54 3.56       869
## 52 3.55       853
## 53 3.56       866
## 72 3.80       969
## 84 3.42       869
## 1  3.54       965
## 20 2.34       754
## 82 3.74       800
## 41 3.44       865
## 86 3.51       865
## 42 3.46       869
## 22 2.98       763
## 69 3.90       962
## 75 2.91       753
## 61 3.69       967
## 21 2.85       762
## 80 3.32       768
## 13 3.80       965
## 16 3.90       967
## 73 2.77       763
## 71 3.93       969
## 11 3.78       966
## 3  3.59       969
## 14 3.88       969
```

```
ggplot(SVM_Training_Data, aes(x = GPA, y = TestScore, colour = Decision)) +
  geom_point(size=3) +
  ggtitle("Admissions")
```



```
## Train the NN
## https://www.rdocumentation.org/packages/nnet/versions/7.3-12/topics/nnet
MyNN <- nnet(Decision ~ GPA+TestScore, data=SVM_Training_Data,
             size=10, ## 10 nodes in hidden layers
             decay=1.0e-2, ## Changing this can affect results
             maxit=100,
             linout=TRUE)
```

```
## # weights:  63
## initial  value 72.207945
## iter  10 value 65.707040
## iter  20 value 65.535609
## iter  30 value 64.879429
## iter  40 value 35.336741
## iter  50 value 29.949016
## iter  60 value 29.251050
## iter  70 value 28.324453
## iter  80 value 27.404438
## iter  90 value 24.299713
## iter 100 value 22.806747
## final  value 22.806747
## stopped after 100 iterations
```

```
MyNN  # 2-2-3: 2 inputs (GPA and TestScore), 2 hidden layers, and 3 outputs
```

```
## a 2-10-3 network with 63 weights
## inputs: GPA TestScore
## output(s): Decision
## options were - softmax modelling  decay=0.01
```

```
(Prediction<-predict(MyNN, SVM_Test_Data_noLabel, type="class"))
```

```
##  [1] "Admit"    "Waitlist" "Waitlist" "Waitlist" "Admit"    "Waitlist"
##  [7] "Admit"    "Waitlist" "Decline"  "Waitlist" "Waitlist" "Waitlist"
## [13] "Decline"  "Admit"    "Decline"  "Admit"    "Decline"  "Decline"
## [19] "Admit"    "Admit"    "Waitlist" "Admit"    "Admit"    "Admit"
## [25] "Admit"
```

```
table(Prediction, MyTestLabels)
```

```
##           MyTestLabels
## Prediction Admit Decline Waitlist
##   Admit       11       0        0
##   Decline      0       5        0
##   Waitlist     0       2        7
```