ABI Final

Importing the Data

```
#Importing the required libraries
library(readxl)

#Importing the data
MRI_data <- read.csv("C:/Users/Saurabh/Desktop/Home Work/ABI/Group
Work/Final/oasis_longitudinal.csv")

#Printing the summary of the data
summary(MRI_data)

##      Subject.ID              MRI.ID             Group          Visit
##  OAS2_0048:  5   OAS2_0001_MR1:  1   Converted  : 37   Min.   :1.000
##  OAS2_0070:  5   OAS2_0001_MR2:  1   Demented   :146   1st Qu.:1.000
##  OAS2_0073:  5   OAS2_0002_MR1:  1   Nondemented:190   Median :2.000
##  OAS2_0127:  5   OAS2_0002_MR2:  1                     Mean   :1.882
##  OAS2_0017:  4   OAS2_0002_MR3:  1                     3rd Qu.:2.000
##  OAS2_0027:  4   OAS2_0004_MR1:  1                     Max.   :5.000
##  (Other)  :345   (Other)      :367
##     MR.Delay       M.F      Hand          Age             EDUC
##  Min.   :   0.0   F:213   R:373   Min.   :60.00   Min.   : 6.0
##  1st Qu.:   0.0   M:160           1st Qu.:71.00   1st Qu.:12.0
##  Median : 552.0                   Median :77.00   Median :15.0
##  Mean   : 595.1                   Mean   :77.01   Mean   :14.6
##  3rd Qu.: 873.0                   3rd Qu.:82.00   3rd Qu.:16.0
##  Max.   :2639.0                   Max.   :98.00   Max.   :23.0
##
##       SES             MMSE            CDR              eTIV
##  Min.   :1.00   Min.   : 4.00   Min.   :0.0000   Min.   :1106
##  1st Qu.:2.00   1st Qu.:27.00   1st Qu.:0.0000   1st Qu.:1357
##  Median :2.00   Median :29.00   Median :0.0000   Median :1470
##  Mean   :2.46   Mean   :27.34   Mean   :0.2909   Mean   :1488
##  3rd Qu.:3.00   3rd Qu.:30.00   3rd Qu.:0.5000   3rd Qu.:1597
##  Max.   :5.00   Max.   :30.00   Max.   :2.0000   Max.   :2004
##  NA's   :19     NA's   :2
##       nWBV             ASF
##  Min.   :0.6440   Min.   :0.876
##  1st Qu.:0.7000   1st Qu.:1.099
##  Median :0.7290   Median :1.194
##  Mean   :0.7296   Mean   :1.195
##  3rd Qu.:0.7560   3rd Qu.:1.293
##  Max.   :0.8370   Max.   :1.587
##
```

```
#Installing the required packages
#install.packages("magrittr")

#Loading the required libraries
library(magrittr)
```

As we can see from the summary that there are several NA values present in the dataset.

```
#Printing the structure if the dataset
str(MRI_data)

## 'data.frame':    373 obs. of  15 variables:
##  $ Subject.ID: Factor w/ 150 levels "OAS2_0001","OAS2_0002",..: 1 1 2 2 2
3 3 4 4 4 ...
##  $ MRI.ID    : Factor w/ 373 levels "OAS2_0001_MR1",..: 1 2 3 4 5 6 7 8 9
10 ...
##  $ Group     : Factor w/ 3 levels "Converted","Demented",..: 3 3 2 2 2 3 3
3 3 3 ...
##  $ Visit     : int  1 2 1 2 3 1 2 1 2 3 ...
##  $ MR.Delay  : int  0 457 0 560 1895 0 538 0 1010 1603 ...
##  $ M.F       : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 1 2 2 2 ...
##  $ Hand      : Factor w/ 1 level "R": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Age       : int  87 88 75 76 80 88 90 80 83 85 ...
##  $ EDUC      : int  14 14 12 12 12 18 18 12 12 12 ...
##  $ SES       : int  2 2 NA NA NA 3 3 4 4 4 ...
##  $ MMSE      : int  27 30 23 28 22 28 27 28 29 30 ...
##  $ CDR       : num  0 0 0.5 0.5 0.5 0 0 0 0.5 0 ...
##  $ eTIV      : int  1987 2004 1678 1738 1698 1215 1200 1689 1701 1699 ...
##  $ nWBV      : num  0.696 0.681 0.736 0.713 0.701 0.71 0.718 0.712 0.711
0.705 ...
##  $ ASF       : num  0.883 0.876 1.046 1.01 1.034 ...

#Drop hand
table(MRI_data$Hand)

##
##   R
## 373

#
MRI_data$Hand<-NULL

#Drop Ids
subject_id<-MRI_data$Subject.ID
MRI_id<-MRI_data$MRI.ID

MRI_data$Subject.ID<-NULL
MRI_data$MRI.ID<-NULL
```

```r
#Checking for null values
sort(apply(MRI_data, 2, function(x){sum(is.na(x))}), decreasing = TRUE)
```

```
##      SES     MMSE    Group    Visit MR.Delay      M.F      Age     EDUC
##       19        2        0        0        0        0        0        0
##      CDR     eTIV     nWBV      ASF
##        0        0        0        0
```

```r
#Printing the tabular structure of the SES values
table(MRI_data$SES)
```

```
##
##   1   2   3   4   5
##  88 103  82  74   7
```

After generating various descriptions of the data, we find there to be 371 observations and 15 columns.Two of those columns are ID which are dropped and another two of them measure the whether a patient has Alzheimer's disease and to what degree. Since our goal is to predict whether a patient has Alzheimer's so we collapse CDR into a binary variable which identifies the presence of Alzheimer's or not. In addition, handedness is the same value for all observations which makes it useless.

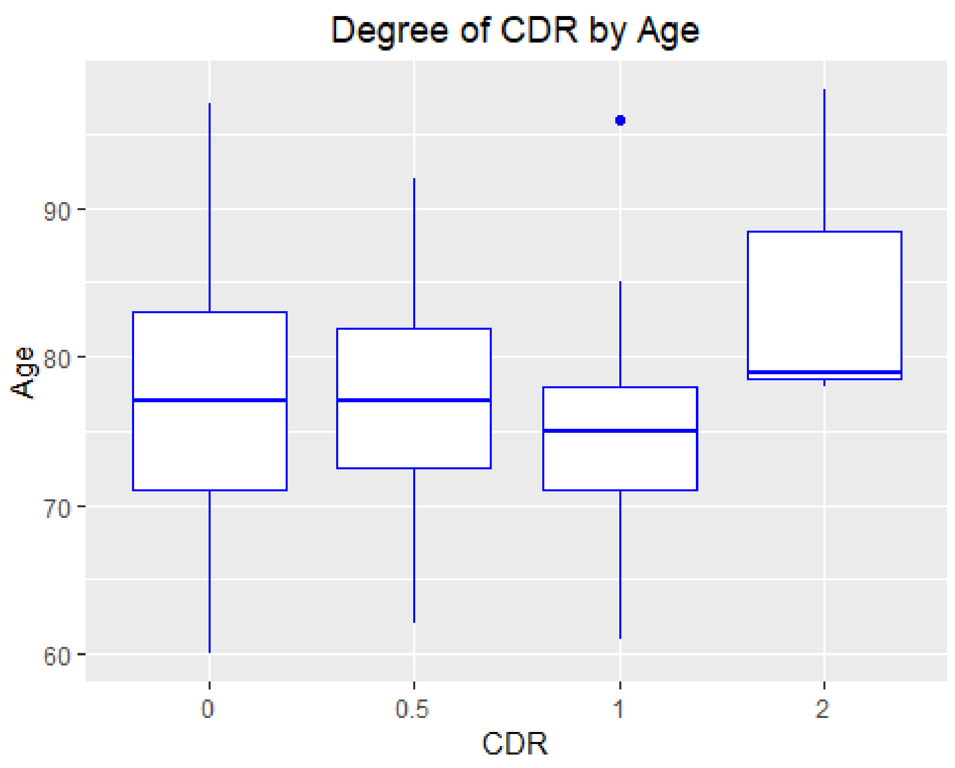
Of the 371 observations, 21 of them have missing data. Given the size of the data set, dropping these observations would lose too much information. Thus we will need to make imputation.

Exploratory Data Analysis and Visualization

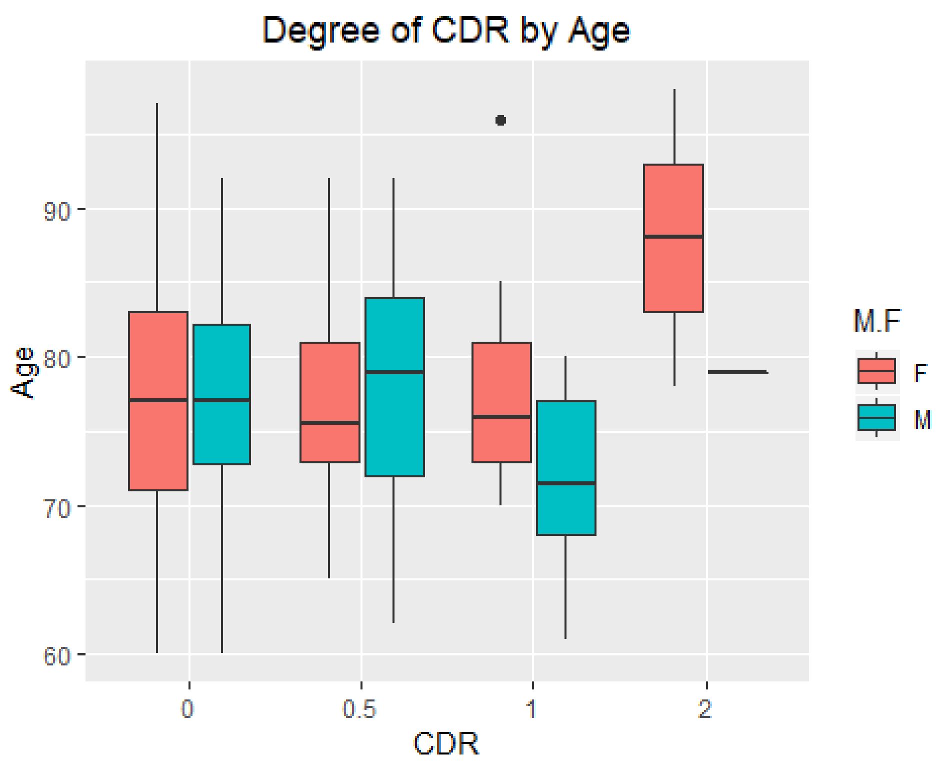
CDR: Clinical Dementia Rating

We explore the relaionship between CDR (CDR: Clinical Dementia Rating) and age

```r
#Importing the required libraries
library(ggplot2)

#Plotting CDR vs age using ggplot
ggplot(MRI_data, aes(as.factor(CDR), Age))+
  geom_boxplot(col = "blue")+
  ggtitle("Degree of CDR by Age")+
  xlab("CDR")+
  theme(plot.title = element_text(hjust = .5))
```



```r
#Plotting CDR against age and using fill as gender
ggplot(MRI_data, aes(as.factor(CDR), Age, fill = M.F))+
  geom_boxplot()+
  ggtitle("Degree of CDR by Age")+
  xlab("CDR")+
  #geom_text(stat = "count", aes(label = ..count..), y = 60, col = "red")+
  theme(plot.title = element_text(hjust = .5))
```



An oddity is the distribution of CDR=2. It is oddly skewed and inconsistent with the pattern. To better understand what is going on, I grouped by Male/Female. We see that there is no variability for Males where CDR=2

Grouping the data based on the CDR rating and gender

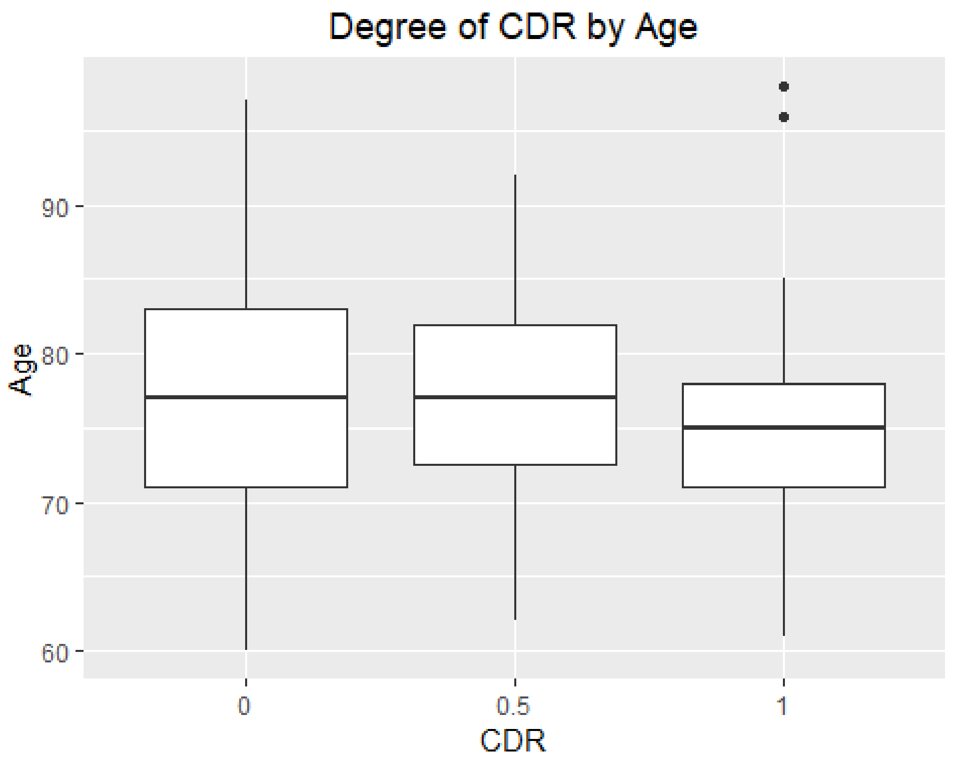
```
#install.packages("dplyr")
library(dplyr)

##
## Attaching package: 'dplyr'

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

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

MRI_data%>%group_by(as.factor(CDR), as.factor(M.F))%>% summarise(n = n())

## # A tibble: 8 x 3
## # Groups:   as.factor(CDR) [?]
##   `as.factor(CDR)` `as.factor(M.F)`     n
##   <fct>            <fct>            <int>
## 1 0                F                  142
## 2 0                M                   64
## 3 0.5                F                   50
## 4 0.5                M                   73
## 5 1                  F                   19
## 6 1                  M                   22
## 7 2                  F                    2
## 8 2                  M                    1
```

Creating a table shows that only 3 observations belong to CDR=2

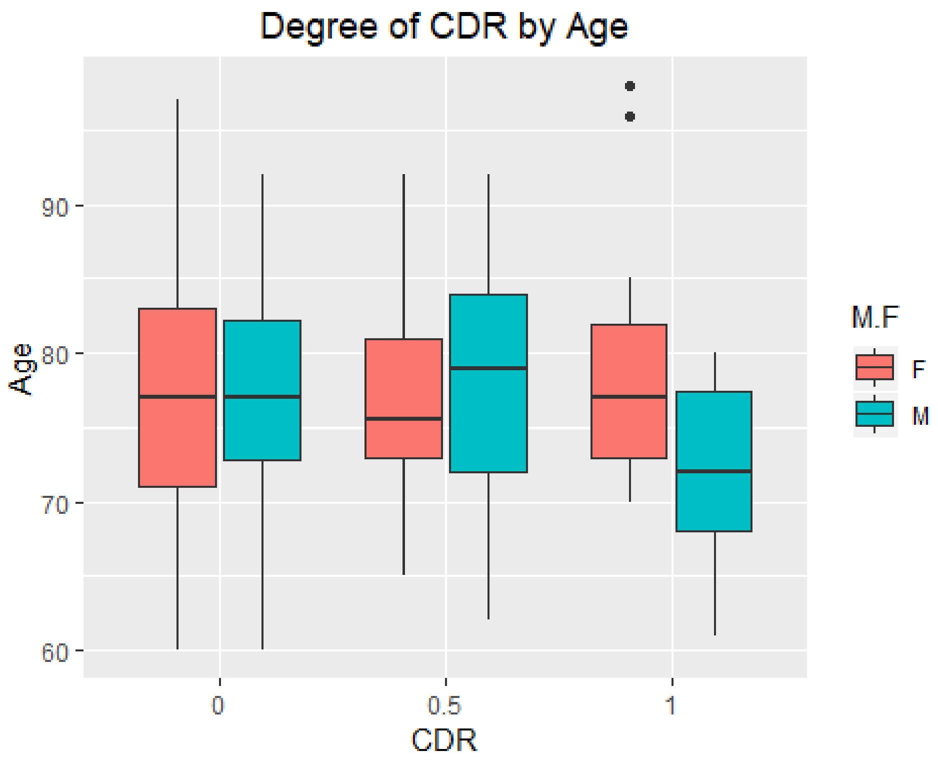
```
#Printing the tabular structure of CD column
table(MRI_data$CD)

## 
##   0 0.5   1   2 
## 206 123  41   3
```

Since we will eventually predict CDR, we are going to group CDR=1 and CDR=2.

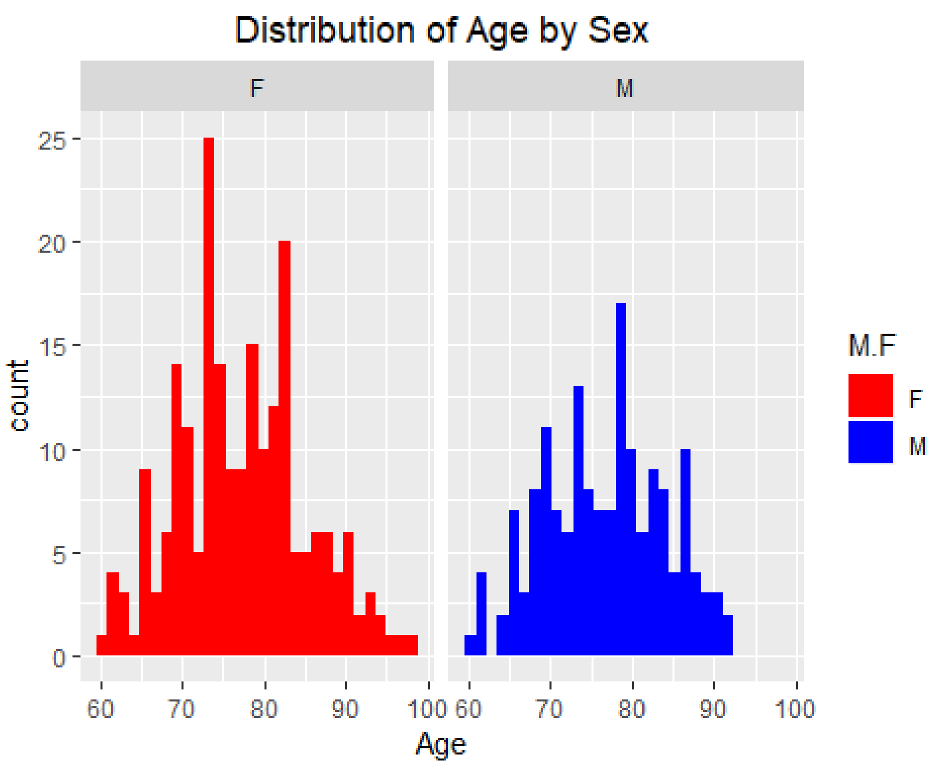
```
#Converting all the CDR values to 1
MRI_data$CDR<-ifelse(MRI_data$CDR==2, 1, MRI_data$CDR)

#Distribution looks better after binning
ggplot(MRI_data, aes(as.factor(CDR), Age))+
  geom_boxplot()+
  ggtitle("Degree of CDR by Age")+
  xlab("CDR")+
  theme(plot.title = element_text(hjust = .5))
```



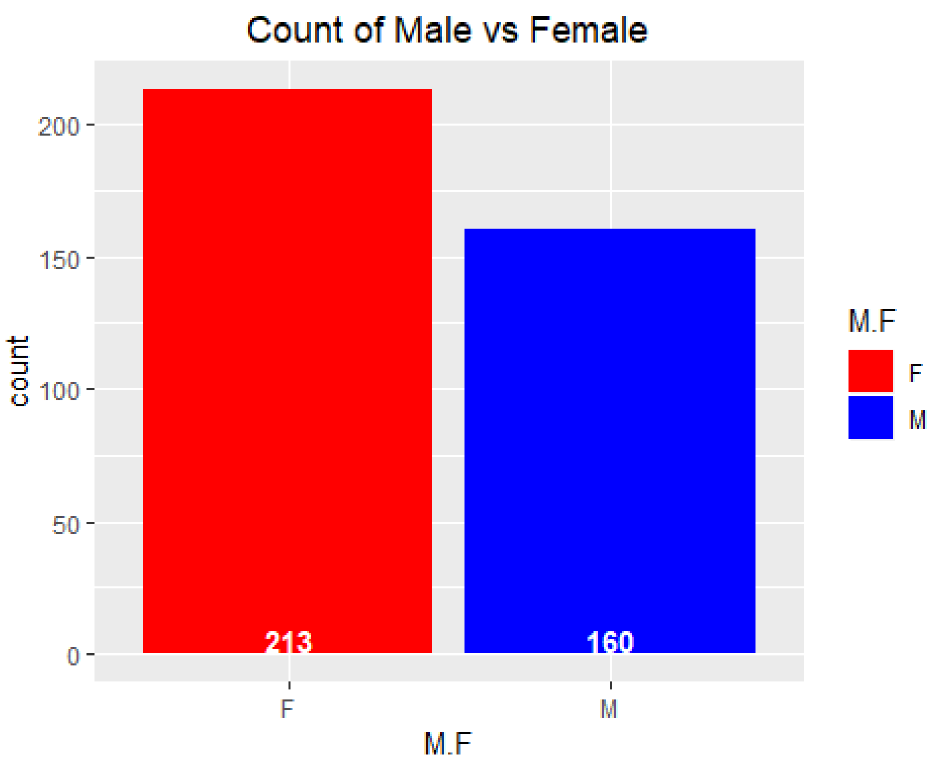
```r
#Plotting a boxplot of CDR against age and for male and female categories
ggplot(MRI_data, aes(as.factor(CDR), Age, fill = M.F))+
  geom_boxplot()+
  ggtitle("Degree of CDR by Age")+
  xlab("CDR")+
  #geom_text(stat = "count", aes(label = ..count..), y = 60, col = "red")+
  theme(plot.title = element_text(hjust = .5))
```



```r
#Plotting a distribution chart of Male and Females for their ages
ggplot(MRI_data, aes(Age, fill = M.F))+
  geom_histogram()+
  facet_wrap(~M.F)+
  scale_fill_manual(values = c("red", "blue"))+
  ggtitle("Distribution of Age by Sex")+
  theme(plot.title = element_text(hjust = .5))

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



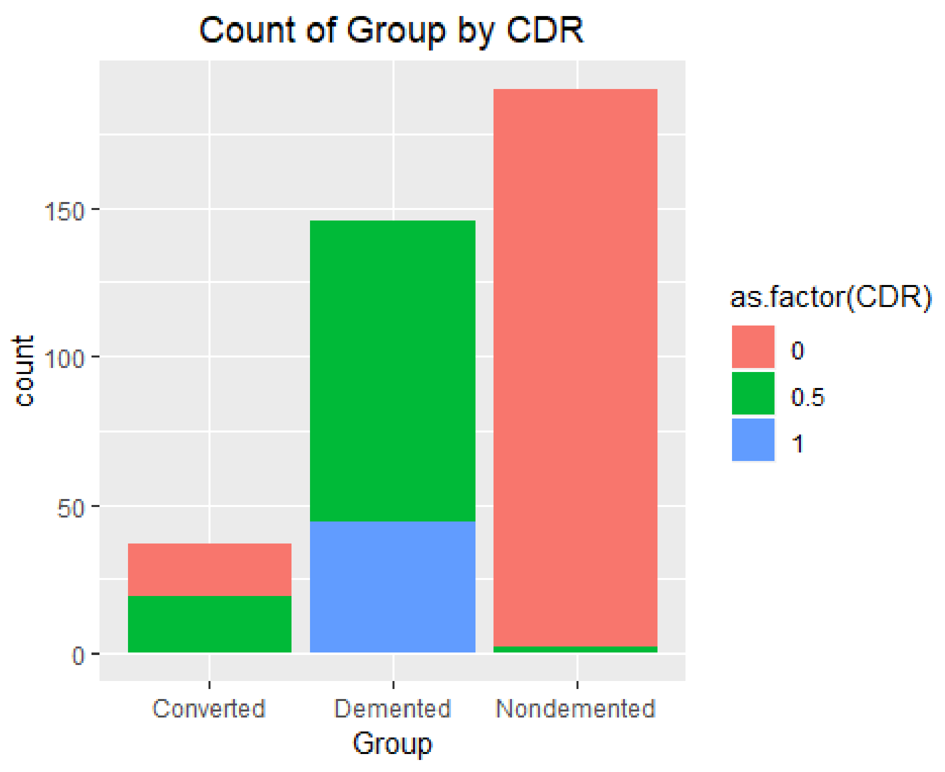
```
#Plotting the barchart for male and female values in the dataset
ggplot(MRI_data, aes(M.F, fill = M.F))+
  geom_bar()+
  scale_fill_manual(values = c("red", "blue"))+
  geom_text(stat = "count", aes(label = ..count..), y = 5, col = "white",
fontface = "bold")+
  ggtitle("Count of Male vs Female")+
  theme(plot.title = element_text(hjust = .5))
```



After binning the CDR values, I think the distributions look a lot better in both cases. When grouping by Male/Female and not.

A logical next step is analyzing Male/Female. I just used this variable in my analysis on CDR and Age, so I should make sure the distributions Male/Female for age are solid. The histograms show that the distributions are almost identical. There is a slight bump in frequency around Age=73 for females which is worth noting. The counts of Male/Female are balanced enough that I shouldnt experience any issue when creating models.

```
#Showing the barchart for Groups as per CDR values
ggplot(MRI_data, aes(Group, fill = as.factor(CDR)))+
  geom_bar()+
  ggtitle("Count of Group by CDR")+
  theme(plot.title = element_text(hjust = .5))
```



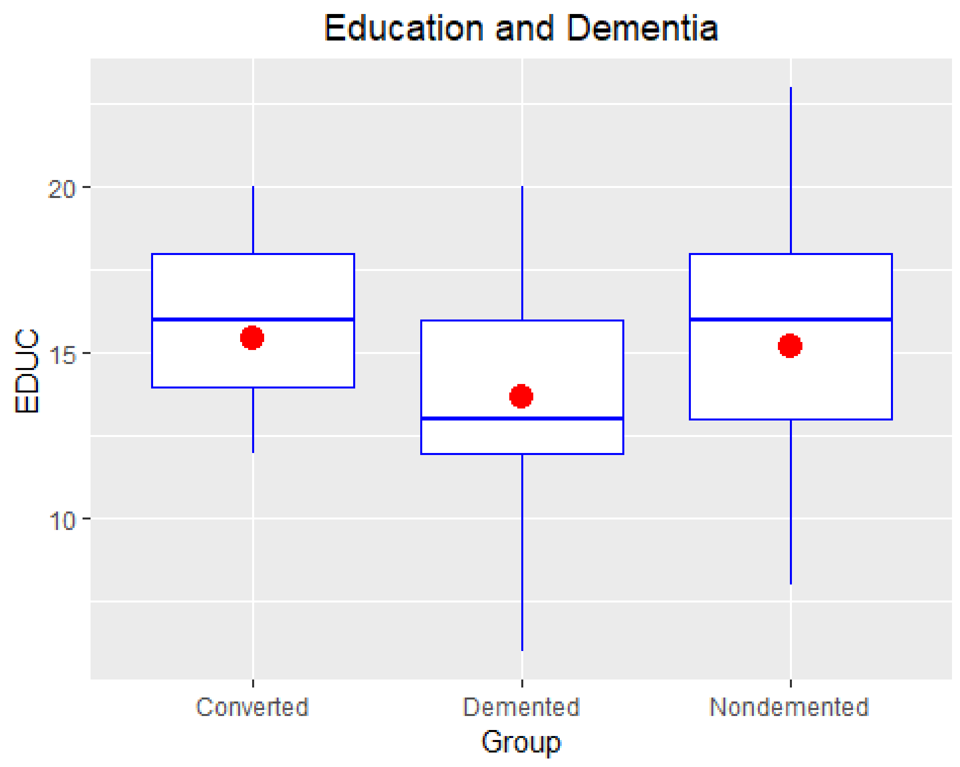
In the Nondemented group, there are 2 observations which have CDR=.5, meaning mild dementia. This does not seem right to me. I wonder if there was a data entry error or something of the sort. we will drop these 2 observations.

We want to further investigate if these variables are similar. For the most part, the stacked bar chart supports our claim that CDR and Group are the same variable

```
#Grouping by Group and CDR to find the count of the records per group
MRI_data%>%group_by(Group, as.factor(CDR))%>%
  summarise(n = n())

## # A tibble: 6 x 3
## # Groups:   Group [?]
##   Group       `as.factor(CDR)`     n
##   <fct>       <fct>            <int>
## 1 Converted   0                   18
## 2 Converted   0.5                 19
## 3 Demented    0.5                102
## 4 Demented    1                   44
## 5 Nondemented 0                  188
## 6 Nondemented 0.5                  2

#get row number
MRI_data[which(MRI_data$Group == "Nondemented" & MRI_data$CDR == .5),]
```

```
##           Group Visit MR.Delay M.F Age EDUC SES MMSE CDR eTIV  nWBV   ASF
## 9  Nondemented     2     1010   M  83   12   4   29 0.5 1701 0.711 1.032
## 31 Nondemented     3      617   M  81   12   3   27 0.5 1814 0.759 0.968
```

```
#Dropping the values of records 9 and 31
MRI_data<-MRI_data[-c(9, 31),]

#Plotting a boxplot of Group for the Education
ggplot(MRI_data, aes(Group, EDUC))+
  geom_boxplot(col = "blue")+
  geom_point(stat = "summary", fun.y = "mean", col = "red", size = 4)+
  ggtitle("Education and Dementia")+
  theme(plot.title = element_text(hjust = .5))
```



Now we explore the relationship between Group and Education. we can say that if perhaps the more education one has, the less likely dementia occurs. There appears to be a difference in median years of education between Demented and Nondemented. As we suspected, the nondemented group has a higher average years of education (red dot) and higher median years of education than Demented.

To further solidify this, I will conduct a Kruskal-Wallis test of medians and an ANOVA of means. Kruskal-Wallis simplifies down to the Wilcoxon Rank Sum test for group=2 and ANOVA simplifies down to a two sample t-test when group=2.

```
#Printing the annova for the aov test
anova(aov(EDUC~M.F, data = MRI_data))
```

```
## Analysis of Variance Table
##
## Response: EDUC
##             Df  Sum Sq Mean Sq F value  Pr(>F)
## M.F          1   27.92 27.9206  3.3933 0.06626 .
## Residuals 369 3036.19  8.2281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Applying the kruskal test for Education
kruskal.test(MRI_data$EDUC, as.factor(MRI_data$M.F))
```

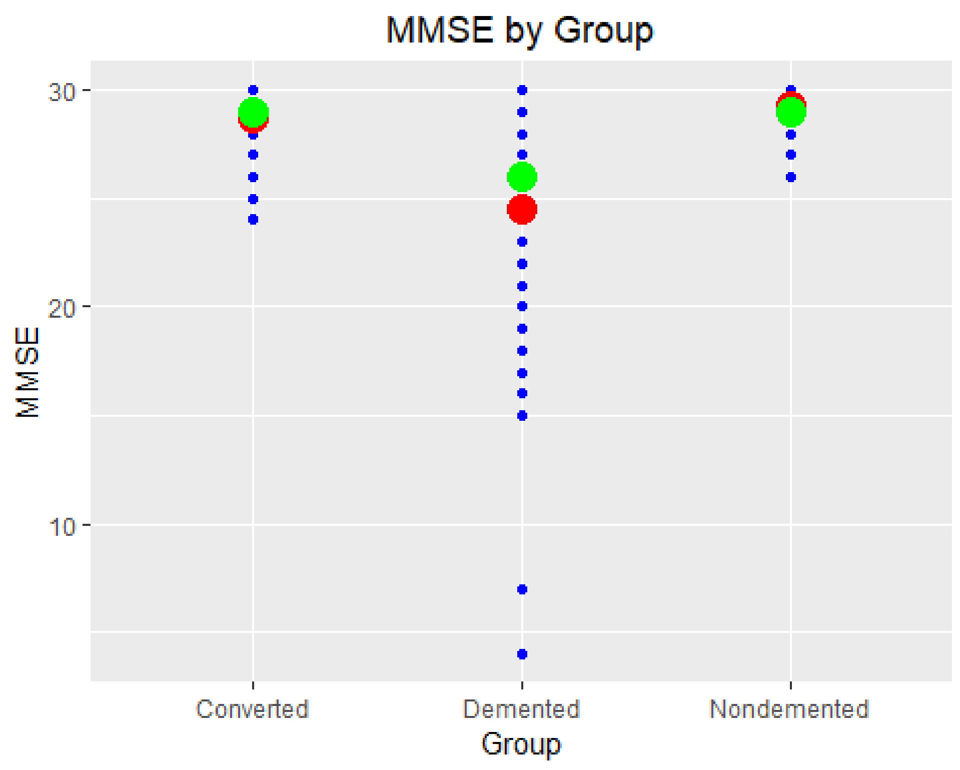
```
##
##  Kruskal-Wallis rank sum test
##
## data:  MRI_data$EDUC and as.factor(MRI_data$M.F)
## Kruskal-Wallis chi-squared = 3.8159, df = 1, p-value = 0.05077
```

The p values for both tests are somewhat low which indicates that there is a difference in medians and means.

MMSE score is usually higher for people without dementia. To verify this we plot a graph to view if its true

```
#There appears to be a slight positive correlation between Education and MMSE score
ggplot(MRI_data, aes(Group, MMSE))+
  geom_point(col = "blue")+
  geom_point(stat = "summary", fun.y = "mean", col = "red", size = 5)+
  geom_point(stat = "summary", fun.y = "median", col = "green", size = 5)+
  ggtitle("MMSE by Group")+
  theme(plot.title = element_text(hjust = .5))
```

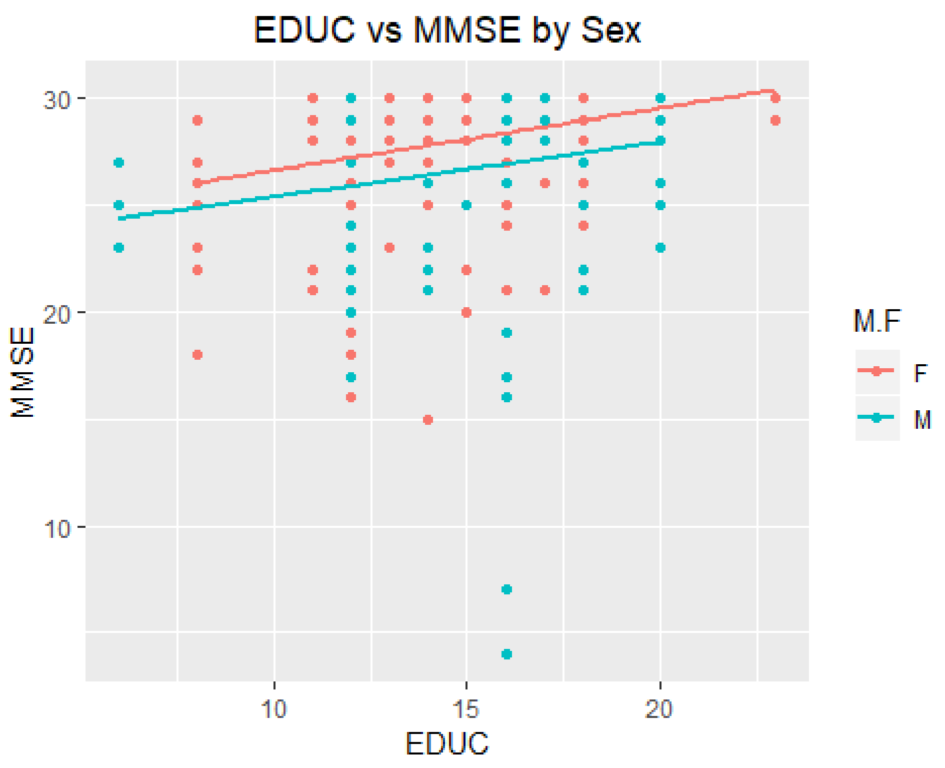
```
## Warning: Removed 2 rows containing non-finite values (stat_summary).

## Warning: Removed 2 rows containing non-finite values (stat_summary).

## Warning: Removed 2 rows containing missing values (geom_point).
```



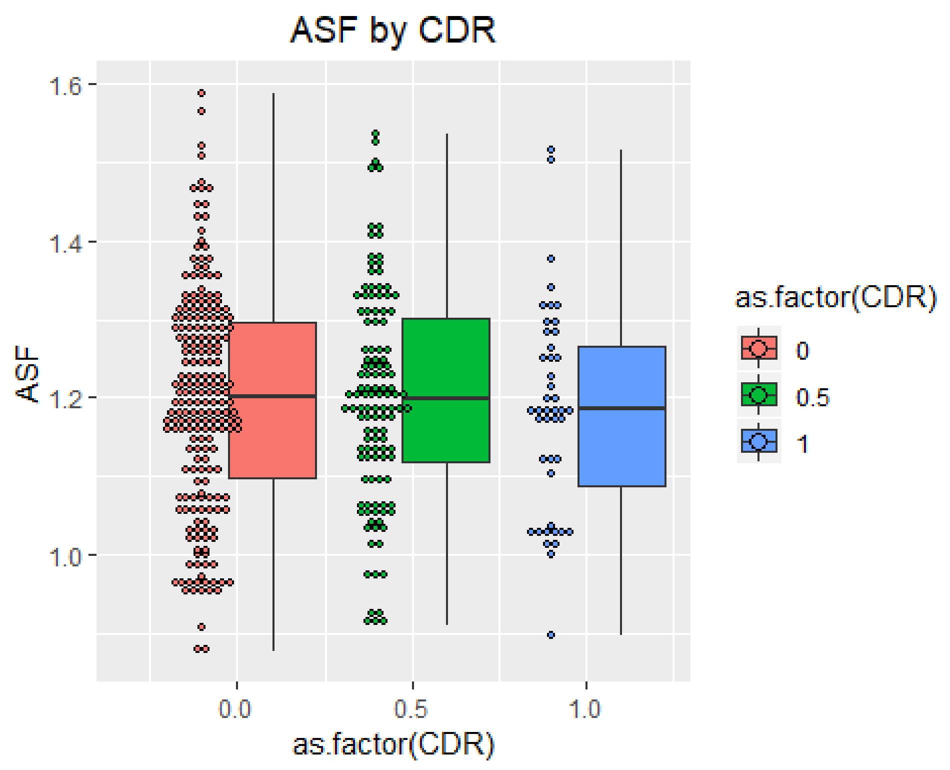
The dot plot with mean (red) and median (green) for each group show that the nondemented group has a higher average and higher median MMSE score.

We want to see if we can identify individuals who score better on MMSE. Our guess is that years of education has some impact. The scatter plot with linear regression lines for Male and Female show a positive correlation between EDUC and MMSE. As suspected, it tends to be that the more years of education one has, the higher the MMSE score. Note that a relationship does not imply causation.

```
#Plotting Education vs MMSE by sex
ggplot(MRI_data, aes(EDUC, MMSE, col = M.F))+
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("EDUC vs MMSE by Sex")+
  theme(plot.title = element_text(hjust = .5))

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Warning: Removed 2 rows containing missing values (geom_point).
```



```
#Plotting boxplot and dotplot for CDR against ASF
ggplot(MRI_data, aes(as.factor(CDR), ASF, fill = as.factor(CDR)))+
  geom_boxplot(aes(x = CDR+.1, group = as.factor(CDR)), width = .25)+
  ggtitle("ASF by CDR")+
  geom_dotplot(aes(x = CDR-.1, group = as.factor(CDR)), binaxis = "y",
binwidth = .01, stackdir = "center")+
  theme(plot.title = element_text(hjust = .5))
```



It appears that the distributions of ASF across different degrees of dementia are very similar. There does appear to be a decrease in variability as the severity of dementia increases.

MODELING

Imputation

As covered earlier, there are some missing values that need to be taken care of. Imputing the median for MMSE is straightforward. There are only two two missing values so there is not much to think about here.

```
#Sorting the data as per NA values
sort(apply(MRI_data, 2, function(x){sum(is.na(x))}), decreasing = TRUE)

##       SES      MMSE     Group     Visit  MR.Delay       M.F       Age      EDUC
##        19         2         0         0         0         0         0         0
##       CDR      eTIV      nWBV       ASF
##         0         0         0         0
```

Imputing values for SES is a bit tricky. SES is an ordinal variable which gets coded as an integer. However, SES is really categorical in nature.

```
#Storing the median for NA values in MMSE column
MRI_data$MMSE<-ifelse(is.na(MRI_data$MMSE), median(MRI_data$MMSE, na.rm =
TRUE), MRI_data$MMSE)
```

```r
#Storing the median for NA values in SES column
MRI_data$SES<-ifelse(is.na(MRI_data$SES), median(MRI_data$SES, na.rm = TRUE),
MRI_data$SES)
```

Factor Variables

Dummy Variables

```r
#install.packages("caret")
library(caret)

## Loading required package: lattice

#Storing the data as factors for relevany columns
MRI_data$Group<-as.factor(MRI_data$Group)
MRI_data$M.F<-as.factor(MRI_data$M.F)
MRI_data$CDR<-as.factor(ifelse(MRI_data$CDR==.5, 1, MRI_data$CDR))

#Storing the encoding in to the variables and then binding those
factor_variables<-MRI_data[, sapply(MRI_data, is.factor)]
CDR<-factor_variables$CDR
dum_var<-dummyVars(~., factor_variables[,-3])
factor_variables<-as.data.frame(predict(dum_var, factor_variables[,-3]))
factor_variables<-as.data.frame(cbind(factor_variables, CDR))
```

We convert appropriate variables to factor and collapse CDR into demented and non demented. To use these the factor variables in the models, we have to turn them into dummy variables. The caret package has a dummyVars function which makes creating dummy variables simple.

Numeric Variables

Multicolinearity

After dealing with factor variables, we turn the attention to numeric variables. It is important to check if any predictors are correlated with each other. If the are, predictive performance may suffer and numerical instability is introduced. The caret package has a function findCorrelation which does a great job of dealing with this issue. We usually pick .75 as a starting point and adjust based on the data. Filtering for correlated predictors drops two variables from the model.

```r
numeric_variables<-MRI_data[, sapply(MRI_data, is.numeric)]

#Calculating the correlation between numeric values and then dropping the
values with highest correlation
correlations<-cor(numeric_variables)
highCorr<-findCorrelation(correlations, cutoff = .75)
numeric_variables<-numeric_variables[,-highCorr]
```

Center and Scale

Next we center and scale all numeric predictors. Some models benefit from having variables on the same scale. We could use the preProcess function in caret, however since there are so few variables, we will just do the center and scaling manually.

```
#Scaling all the numeric values
numeric_variables$Visit<-scale(numeric_variables$Visit)
numeric_variables$Age<-scale(numeric_variables$Age)
numeric_variables$EDUC<-scale(numeric_variables$EDUC)
numeric_variables$MMSE<-scale(numeric_variables$MMSE)
numeric_variables$nWBV<-scale(numeric_variables$nWBV)
numeric_variables$ASF<-scale(numeric_variables$ASF)
```

Dropping group variables.

```
#Creating the training data
train1<-as.data.frame(cbind(numeric_variables, factor_variables))

#Dropping all the irrelevant columns
train1<-train1[,c(-8,-9,-10)]
```

Logistic Regression with L1Penalty

We start by setting the seed to make my results reproducible. We use the glmnet method to fit a logistic regression model with the L1 penalty. This penalty can shrink some coefficients to 0.

```
#lasso
#install.packages("glmnet")
library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16

#Setting the crossvalidation parameter
ctrl<-trainControl(method = "cv", number = 10)

#Setting the seed to 11
set.seed(11)

#Applying the model and applying the crossvalidation as per the parameter
which we set above
lasso.mod<-train(CDR~., data = train1, method = "glmnet", metric =
"Accuracy", trControl = ctrl,tuneGrid = expand.grid(alpha = 1, lambda =
seq(0, .015, length = 30)))

#Printing all the relevant parameters for the model
max(lasso.mod$results$Accuracy)
```

```
## [1] 0.8254188
```

```
lasso.mod$bestTune
```

```
##   alpha      lambda
## 4     1 0.001551724
```

```
coef(lasso.mod$finalModel, lasso.mod$finalModel$lambdaOpt)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept)  2.684385e+00
## Visit        9.392815e-02
## Age         -8.337133e-01
## EDUC        -7.422965e-01
## SES         -4.867299e-01
## MMSE        -3.641442e+00
## nWBV        -9.713014e-01
## ASF          5.497474e-01
## M.F.F       -1.566154e+00
## M.F.M        1.528760e-14
```

This conducts feature selection and helps with overfitting. The best tune was when Î»=.000517 which produces an accuracy of 83%. Based on the coefficients, there werent any variables dropped from the model. It seems that Age, EDUC, SES, and ASF have the largest coefficients. It will be interesting to check the variable importance of other models I run.

Support Vector Machine

Next we fit a support vector machine. These models tend to be powerful at the expense of interpretability.

```
#SVM
set.seed(11)

#Applying SVM model
svm.mod<-train(CDR~., data = train1, method = "svmRadial", metric =
"Accuracy", trControl = ctrl, tuneLength = 15)
varImp(svm.mod)
```

```
## ROC curve variable importance
##
##       Importance
## MMSE    100.0000
## nWBV     52.4295
## M.F.F    34.8586
## M.F.M    34.8586
## EDUC     32.6727
## SES      12.3204
## Age       2.5732
## Visit     0.7786
## ASF       0.0000

#Printing all the relevant parameters for the model
max(svm.mod$results$Accuracy)

## [1] 0.8442745

svm.mod$bestTune

##        sigma  C
## 7 0.09274474 16
```

The best tune is Cost=8 with accuracy of 86%. Notably better than the logistic regression model previously fitted. According to variable importance of the support vector machine, MMSE is by far the most important variable. Second most important is nWBV. Then, sex and EDUC are tied for third. Interesting to compare with the coefficients of the logistic regression.

KNN

Now we create a KNN model.

```
set.seed(11)

#Applying the model as per the CV parameters we set above
knn.mod<-train(CDR~., data = train1, method = "knn", metric = "Accuracy",
trControl = ctrl,tuneGrid = data.frame(.k = 1:20))

#Printing all the relevant columns for the model
knn.mod$bestTune

##   k
## 1 5

max(knn.mod$results$Accuracy)

## [1] 0.8194326
```

The optimal tune is with k=5. This means that when predicting a new point, the five closest points determine what the new one will be. With an accuracy of 81%, there is a drop in performance compared to logistic regression and support vector machine.

Random Forest

Further investigating variable importance, we turn to random forest. we make use of the variable importance aspect of the model to compare with logistic regression support vector machine. Due to the biased nature of randomForest default variable importance method, we use importance = T and type = 1.

```
#install.packages("randomForest")
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

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

set.seed(11)

#Applying the model
rf.mod<-randomForest(CDR~., data = train1, importance =T)
importance(rf.mod, type = 1)

##       MeanDecreaseAccuracy
## Visit            -5.226894
## Age              15.434347
## EDUC             23.197152
## SES              14.072956
## MMSE             75.394738
## nWBV             24.072400
## ASF              17.489972
## M.F.F            13.287232
## M.F.M            12.844535
```

Seems like MMSE is most important, followed by nWBV and EDUC. Very similar to the support vectors variable importance! Interesting! The best tune is listed above and has accuracy of 86%.

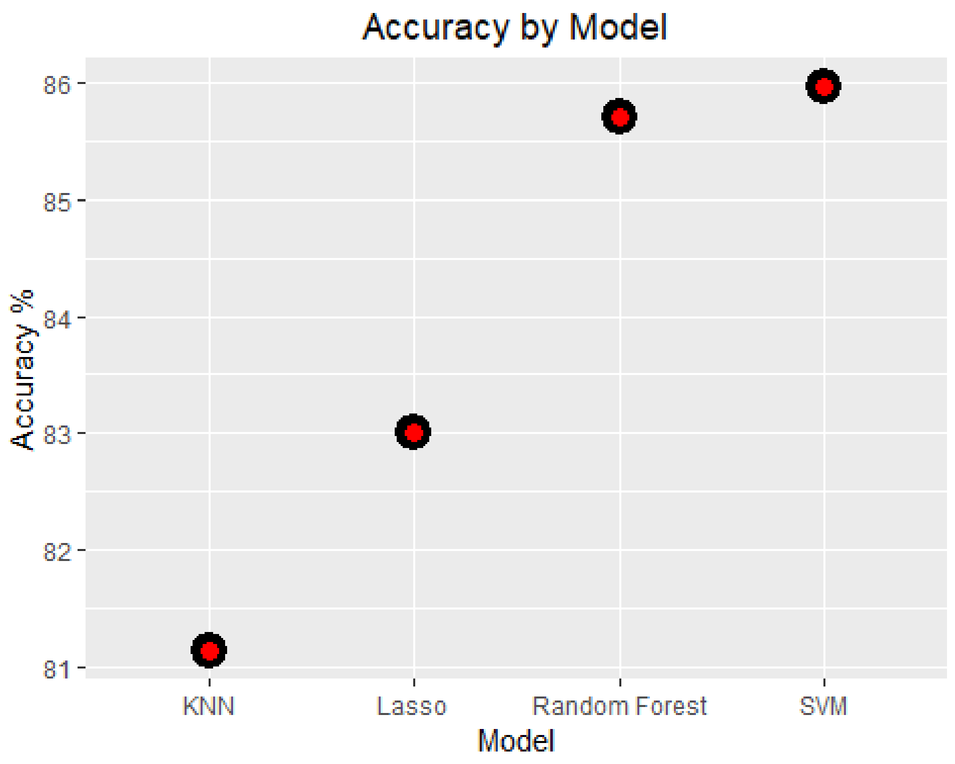
Model Assessment

After creating models, we check their correlation with each other.

```
#Checking the correlation between the models
models<-list("lasso" = lasso.mod, "svm" = svm.mod, "knn" = knn.mod)
modelCor(resamples(models))

##           lasso       svm       knn
## lasso 1.0000000 0.4491280 0.1936942
## svm   0.4491280 1.0000000 0.1759831
## knn   0.1936942 0.1759831 1.0000000
```

Looks like lasso is most correlated with xgb and not very correlated with the support vector machine and KNN. Support vector is most correlated with Random Forest, but, .57 is not that high of a correlation. Finally KNN is not strongly correlated with any of the other models. If we were to try stacking or some other type of ensemble, these models would be good candidates since they are fit very differently and have low correlation.

```r
#Printing the model accuracy
mod_accuracy<-c(0.8299826, 0.8596412,
                0.8570887, 0.8113482)
Model<-c("Lasso", "SVM", "Random Forest", "KNN")

#Plotting the model accuracy for models
data.frame(Model, mod_accuracy)%>%
  ggplot(aes(reorder(Model, mod_accuracy), mod_accuracy*100))+
  geom_point(col = "black", size = 6)+
  geom_point(col = "red", size = 3)+
  ggtitle("Accuracy by Model")+
  ylab("Accuracy %")+
  xlab("Model")+
  theme(plot.title = element_text(hjust = .5))
```



In terms of individual model performance based on accuracy, we see that the support vector performs the best with Random Forest close behind. There is a noticeable drop to lasso and another big drop to KNN.