STOR 765 Project - MME & Fall

Yali Li

3/29/2021

Part 0 - Data input and manipulate

```r
rm(list=ls())

library(openxlsx)
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
```

```r
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 3.0-2
```

```r
library(ROCR)
```

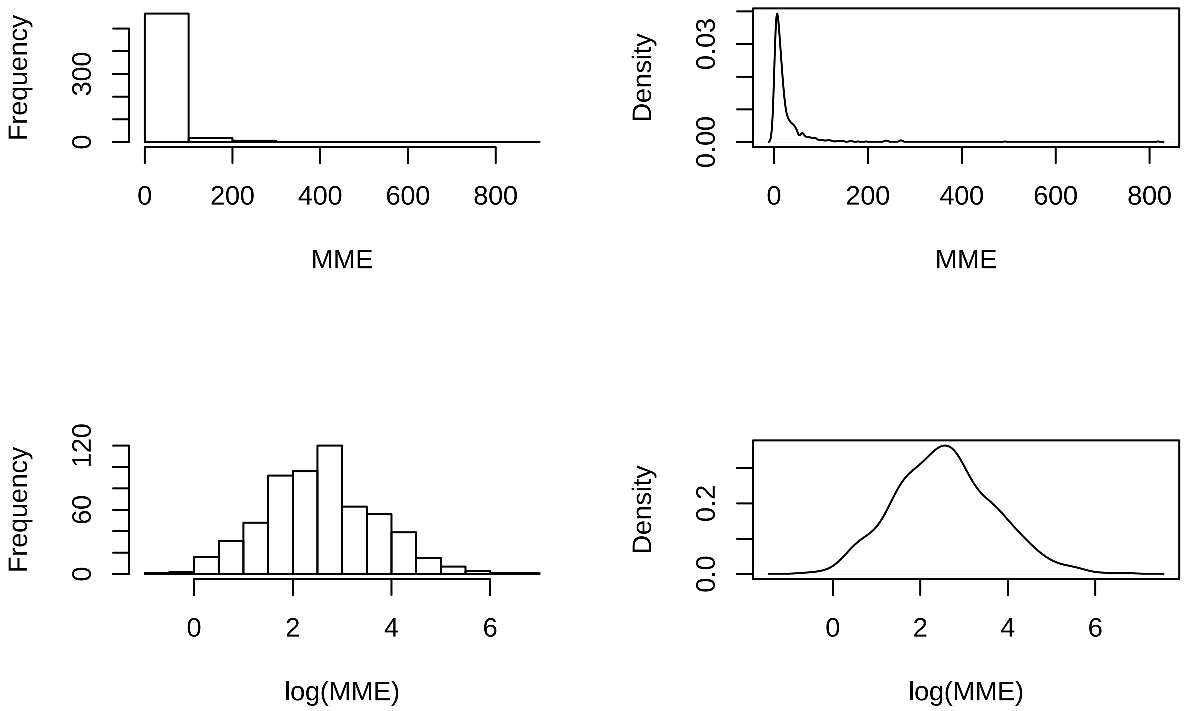
```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(caTools)

setwd("/Users/yali/UNC/STOR 765/MME & Fall")

# Input data
mmedata.raw <- read.xlsx("Dataset 02152021 Phase 1 and Phase 2.xlsx")

# Remove missing value in STEADI_q_yn and Cancer_Indicator=1
# mmedata <- filter(mmedata.raw, STEADI_q_yn !="." & Cancer_Indicator == 0)

# Remove missing value in STEADI_q_yn
mmedata <- filter(mmedata.raw, STEADI_q_yn !=".")

# Drop some columns
mmedata <- select(mmedata, -Patient_ID,-Phase)

# replace '.' with NA
mmedata$Acute_Pain_Indicator = na_if(mmedata$Acute_Pain_Indicator,'.')
mmedata$COPD_Indicator = na_if(mmedata$COPD_Indicator,'.')
mmedata$CVD_Indicator = na_if(mmedata$CVD_Indicator,'.')
mmedata$Sleep_Apnea_Indicator = na_if(mmedata$Sleep_Apnea_Indicator,'.')
mmedata$Cancer_Indicator = na_if(mmedata$Cancer_Indicator,'.')
mmedata$Psychiatric_Indicator = na_if(mmedata$Psychiatric_Indicator,'.')

# Convert the data type to numeric or factor
mmedata$STEADI_q_yn <- as.numeric(mmedata$STEADI_q_yn)
mmedata$Acute_Pain_Indicator <- as.numeric(mmedata$Acute_Pain_Indicator)
mmedata$COPD_Indicator <- as.numeric(mmedata$COPD_Indicator)
mmedata$CVD_Indicator <- as.numeric(mmedata$CVD_Indicator)
mmedata$Sleep_Apnea_Indicator <- as.numeric(mmedata$Sleep_Apnea_Indicator)
mmedata$Cancer_Indicator <- as.numeric(mmedata$Cancer_Indicator)
mmedata$Psychiatric_Indicator <- as.numeric(mmedata$Psychiatric_Indicator)

mmedata$Patient_Gender <- as.factor(mmedata$Patient_Gender)
mmedata$Patient_Race <- as.factor(mmedata$Patient_Race)
```

Part 1

(a) Histgram and Kernel Density Estimation of MME

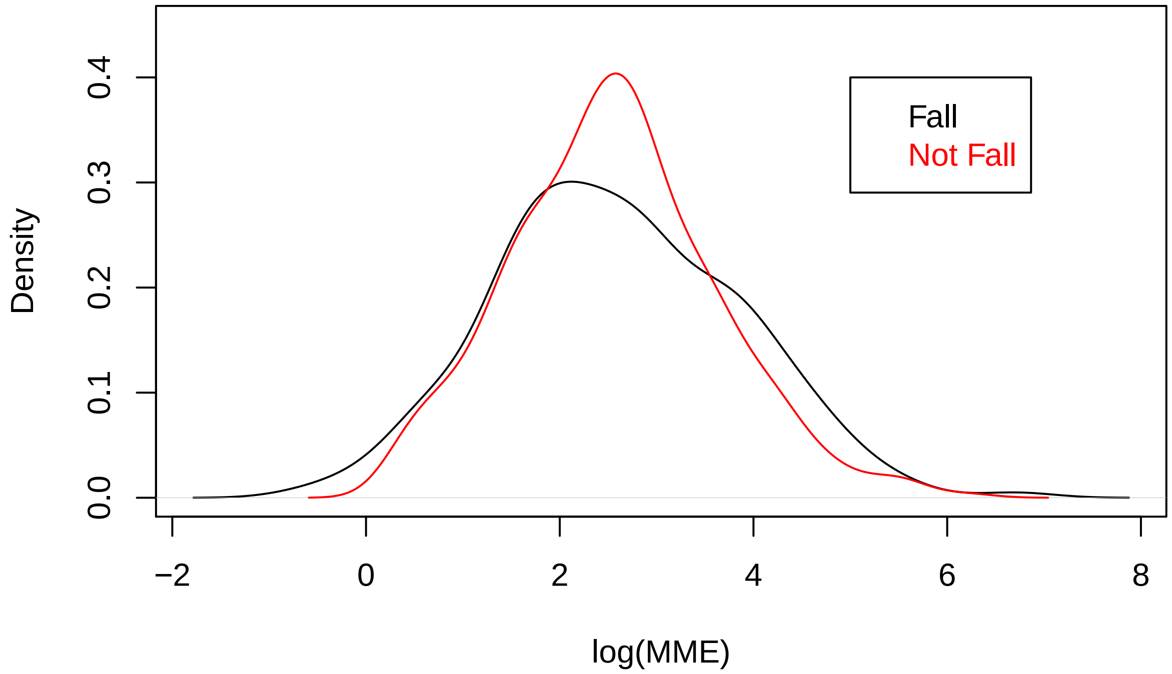
```r
daily.exp <- mmedata$avg_daily_exp_fl_lkbk
steadi <- mmedata$STEADI_q_yn
age <- mmedata$Patient_Age_at_Visit_Date

par(mfrow=c(2,2))
hist(daily.exp, xlab = "MME", main="")
plot(density(daily.exp),xlab = "MME", main="",col=1)

hist(log(daily.exp), xlab = "log(MME)", main="")
plot(density(log(daily.exp)),xlab = "log(MME)", main="",col=1)
```



> log(MME) is approximately normally distributed

(b) Compare the MME distribution in groups of fall and not fall

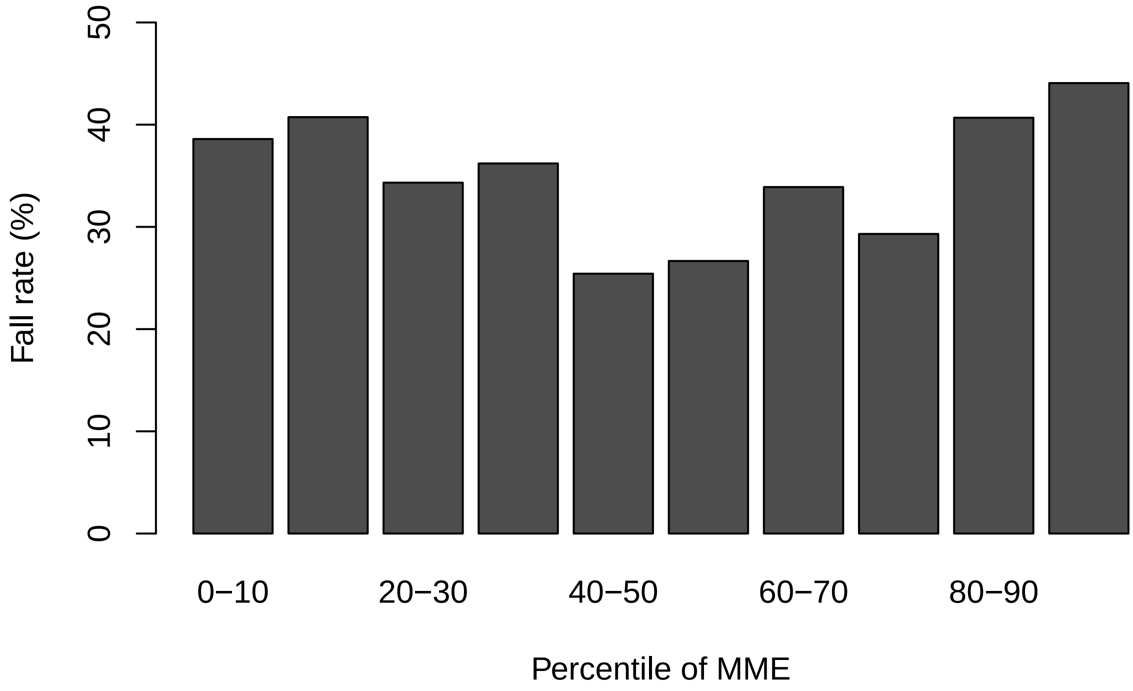
```r
daily.exp_fall <- mmedata$avg_daily_exp_fl_lkbk[mmedata$STEADI_q_yn == 1]
daily.exp_notfall <- mmedata$avg_daily_exp_fl_lkbk[mmedata$STEADI_q_yn == 0]

age_fall <- mmedata$Patient_Age_at_Visit_Date[mmedata$STEADI_q_yn == 1]
age_notfall <- mmedata$Patient_Age_at_Visit_Date[mmedata$STEADI_q_yn == 0]

# density
par(mfrow=c(1,1))

den.fall <- density(log(daily.exp_fall))
plot(den.fall,xlab = "log(MME)", main="", col=1, ylim=c(0,0.45))
# abline(v=den.fall$x[which.max(den.fall$y)],col=1,lty=2)

den.notfall <- density(log(daily.exp_notfall))
lines(den.notfall,xlab = "log(MME)",main="Not Fall",col=2)
# abline(v=den.notfall$x[which.max(den.notfall$y)],col=2,lty=2)
legend(5,0.4,c('Fall','Not Fall'),text.col=c(1,2))
```



> The MME distributions are similar for the groups of fall and not fall. The MME distribution of the fall group skewed to the left slightly.

(c) barplot: MME vs Fall Rate

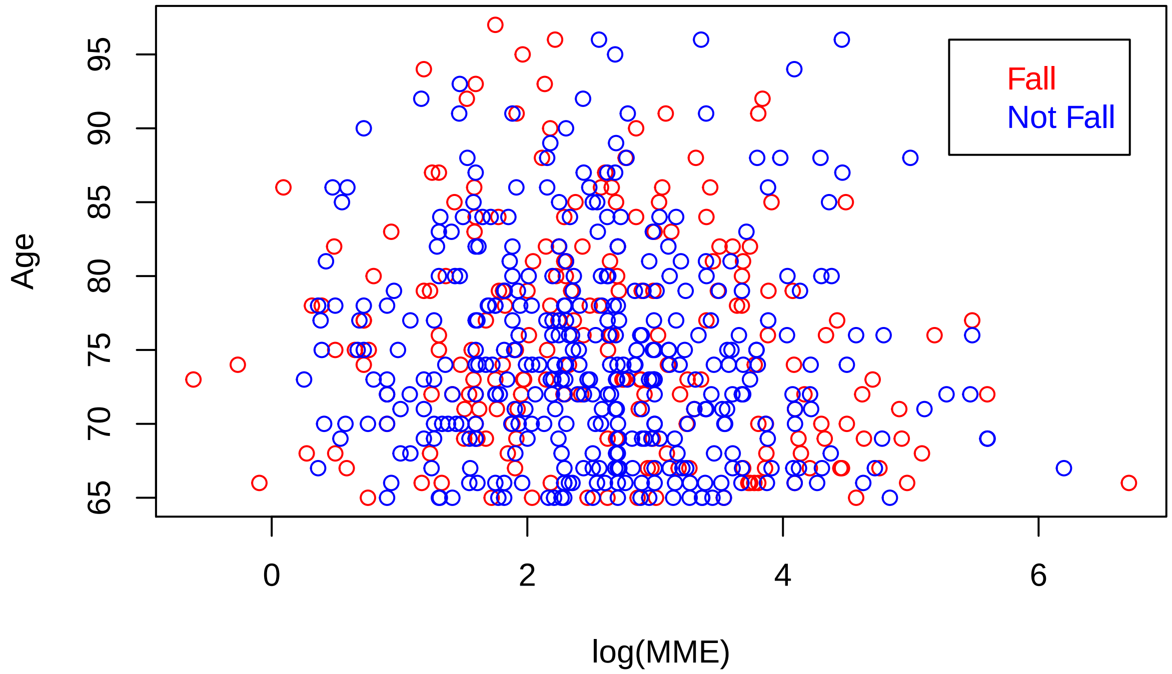
```
daily.exp_quan <- quantile(daily.exp, c(seq(0,1,0.1)))
fall.rate <- matrix(data = NA, nrow =length(daily.exp_quan)-1, ncol = 1)

for (i in 1:length(daily.exp_quan)-1) {
  fall.rate[i]<- mean(steadi[daily.exp>=daily.exp_quan[i] & daily.exp<daily.exp_quan[i+1]])*100
}

barplot(height=t(fall.rate),names.arg=c("0-10","10-20","20-30","30-40","40-50","50-60","60-70","70-80",
```



> No obivious trend was found in the fall rate (number of peopel fall / total number) when MME increases

(d) scatter plot

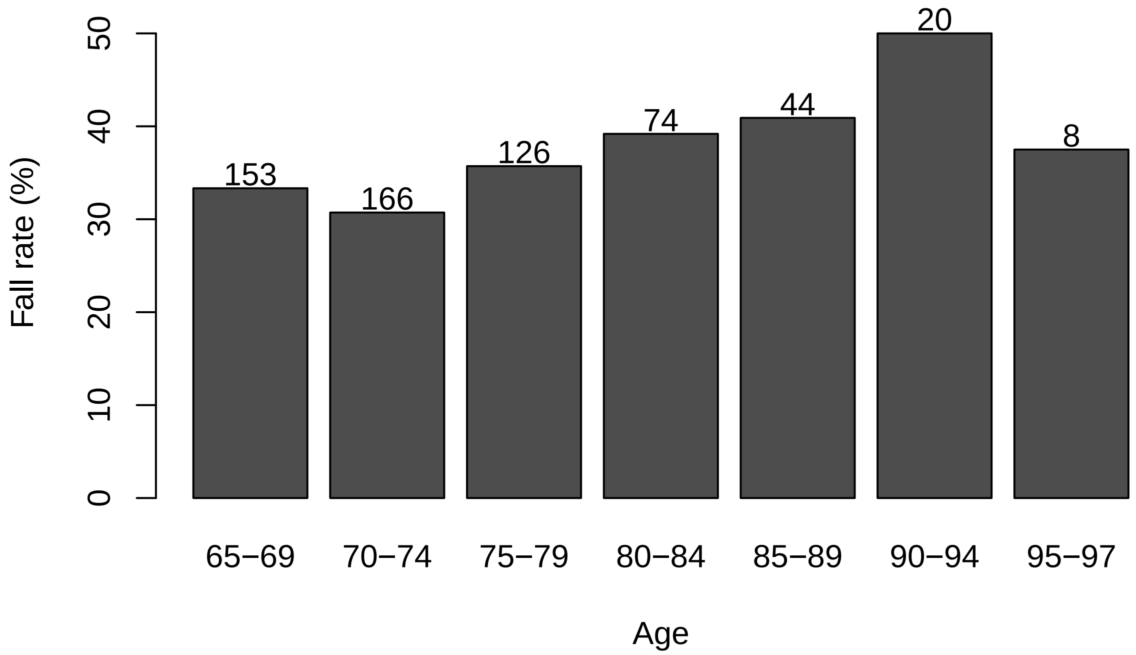
```
plot(log(daily.exp_fall),age_fall,col=2,xlab="log(MME)",ylab="Age")
points(log(daily.exp_notfall),age_notfall,col=4)
legend(5.3,96,c('Fall','Not Fall'),text.col=c(2,4))
```



> The points in the two group don't apart away

(e) barplot: Age vs Fall Rate

```
fall.rate2 <- matrix(data = NA, nrow =7, ncol = 1)
num_of_people <- matrix(data = NA, nrow =7, ncol = 1)
for (i in 1:7) {
  fall.rate2[i]<- mean(steadi[age>=60+5*i & age<60+5*(i+1)])*100
  num_of_people[i]<- length((steadi[age>=60+5*i & age<60+5*(i+1)]))
}

barplot(height=t(fall.rate2),names.arg=c("65-69","70-74","75-79","80-84","85-89","90-94","95-97"),xlab =
text(seq(0.7,8,1.2),fall.rate2+1.5,labels = num_of_people)
```



> Overall, the fall rate increases wiith age. Note the sample number for the group of 95-97 is only 8.

Part 2

Logistic Regression

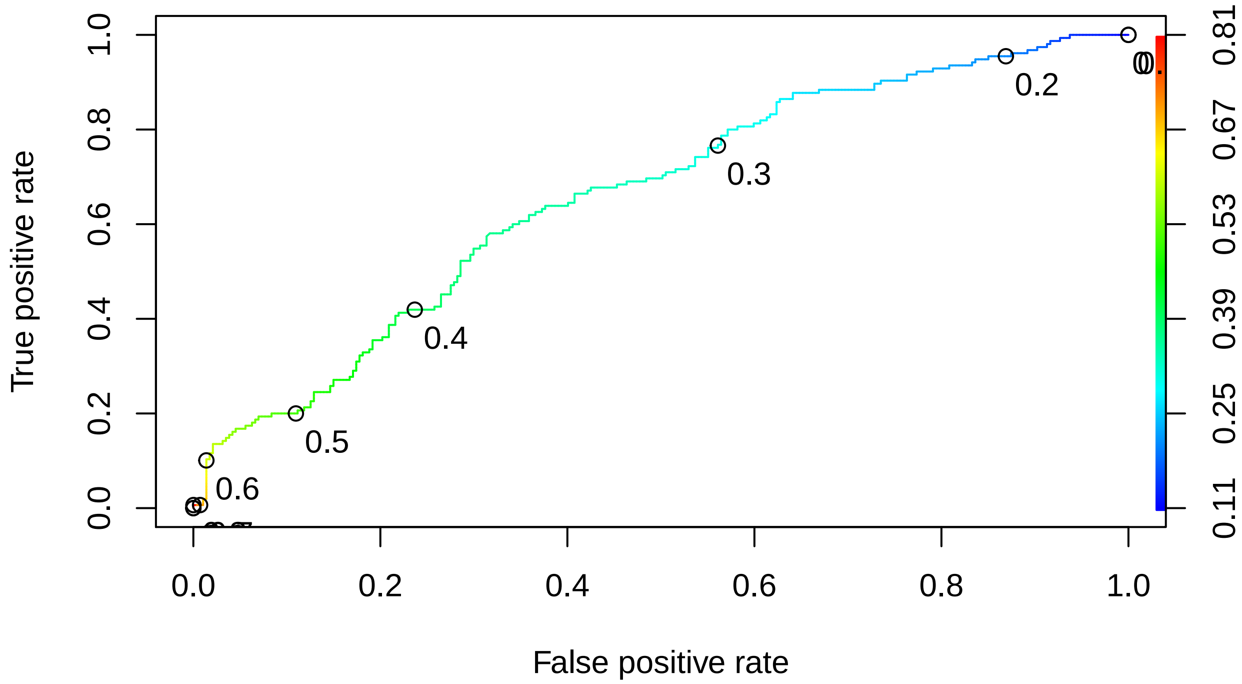
```
mmedata$STEADI_q_yn <- as.factor(mmedata$STEADI_q_yn)
mmedata$Patient_Gender <- as.numeric(mmedata$Patient_Gender)
mmedata$Patient_Race <- as.numeric(mmedata$Patient_Race)

mmedata <- filter(mmedata, !is.na(mmedata$Cancer_Indicator))

# split the whole dataset (n=590) to train (75%) and test (25%) datasets
set.seed(1)
split = sample.split(mmedata$STEADI_q_yn, SplitRatio = 0.75)
mmedata.train = subset(mmedata, split == TRUE)
mmedata.test = subset(mmedata, split == FALSE)

# train the model
# glm.fits = glm (STEADI_q_yn ~ ., family = binomial, data = mmedata.train)
# glm.fits = glm (STEADI_q_yn ~ . -Benzo_indicator -Acute_Pain_Indicator -Patient_Race, family = binomi
glm.fits = glm (STEADI_q_yn ~ .-Benzo_indicator-Acute_Pain_Indicator-Patient_Race-CVD_Indicator, family

summary(glm.fits)
```

```
##
## Call:
## glm(formula = STEADI_q_yn ~ . - Benzo_indicator - Acute_Pain_Indicator -
##     Patient_Race - CVD_Indicator, family = binomial, data = mmedata.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5703  -0.9244  -0.7228   1.2590   1.9862
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.901330   1.218755  -2.381  0.01729 *
## avg_daily_exp_fl_lkbk       0.002696   0.001867   1.444  0.14880
## Patient_Gender             -0.631706   0.230617  -2.739  0.00616 **
## Patient_Age_at_Visit_Date   0.031056   0.014756   2.105  0.03532 *
## COPD_Indicator              0.585725   0.252093   2.323  0.02016 *
## Sleep_Apnea_Indicator       0.520857   0.268841   1.937  0.05269 .
## Cancer_Indicator            0.375539   0.233180   1.611  0.10729
## Psychiatric_Indicator       0.621028   0.223631   2.777  0.00549 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 572.71  on 441  degrees of freedom
## Residual deviance: 539.59  on 434  degrees of freedom
## AIC: 555.59
##
## Number of Fisher Scoring iterations: 4
```

```r
glm.probs = predict(glm.fits,type="response")

# mean predicted probability for the observed not_fall and fall groups
tapply(glm.probs,mmedata.train$STEADI_q_yn,mean)
```

```
##         0         1
## 0.3253500 0.3975777
```

```r
# Get the ROC curve to select a value for the probability threshold
ROCRpred = prediction (glm.probs, mmedata.train$STEADI_q_yn)
ROCRperf = performance(ROCRpred,"tpr","fpr")
plot(ROCRperf,colorize=TRUE,print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))
```



```
predictTest = predict(glm.fits,type="response",newdata = mmedata.test)
ROCRpredTest = prediction(predictTest, mmedata.test$STEADI_q_yn)
auc = as.numeric(performance(ROCRpredTest, "auc")@y.values)
print(auc)
```

```
## [1] 0.6314103
```

```
# Based on the ROC curve, a threshold value of 0.4 is selected to have balanced Sensitivity and Specifi
glm.pred.test=rep(0,length(predictTest))
glm.pred.test[predictTest>0.4]=1

# Confusion Matrix
confusion_mat <- table(mmedata.test$STEADI_q_yn, glm.pred.test)
colnames(confusion_mat) <- c("Predicted.Not_Fall", "Predicted.Fall")
rownames(confusion_mat) <- c("Actual.Not_Fall", "Actual.Fall")
print(confusion_mat)
```

```
##                  glm.pred.test
##                   Predicted.Not_Fall Predicted.Fall
##   Actual.Not_Fall                 73             23
##   Actual.Fall                     26             26
```

```
TN = confusion_mat[1,1]
FP = confusion_mat[1,2]
```

```
FN = confusion_mat[2,1]
TP = confusion_mat[2,2]

paste('Overall accuracy =',round((TN+TP)/(TN+FP+FN+TP),3))
```

```
## [1] "Overall accuracy = 0.669"
```

```
paste('Sensitivity =',round(TP/(TP+FN),3))
```

```
## [1] "Sensitivity = 0.5"
```

```
paste('Specificity =',round(TN/(TN+FP),3))
```

```
## [1] "Specificity = 0.76"
```

A few tests have been done. Finally, a logistic regression model was fitted to predict STEADI_q_yn (fall or not fall) using avg_daily_exp_fl_lkbk, Patient_Gender, Patient_Age_at_Visit_Date, COPD_Indicator, Sleep_Apnea_Indicator,Cancer_Indicator, and Psychiatric_Indicator.

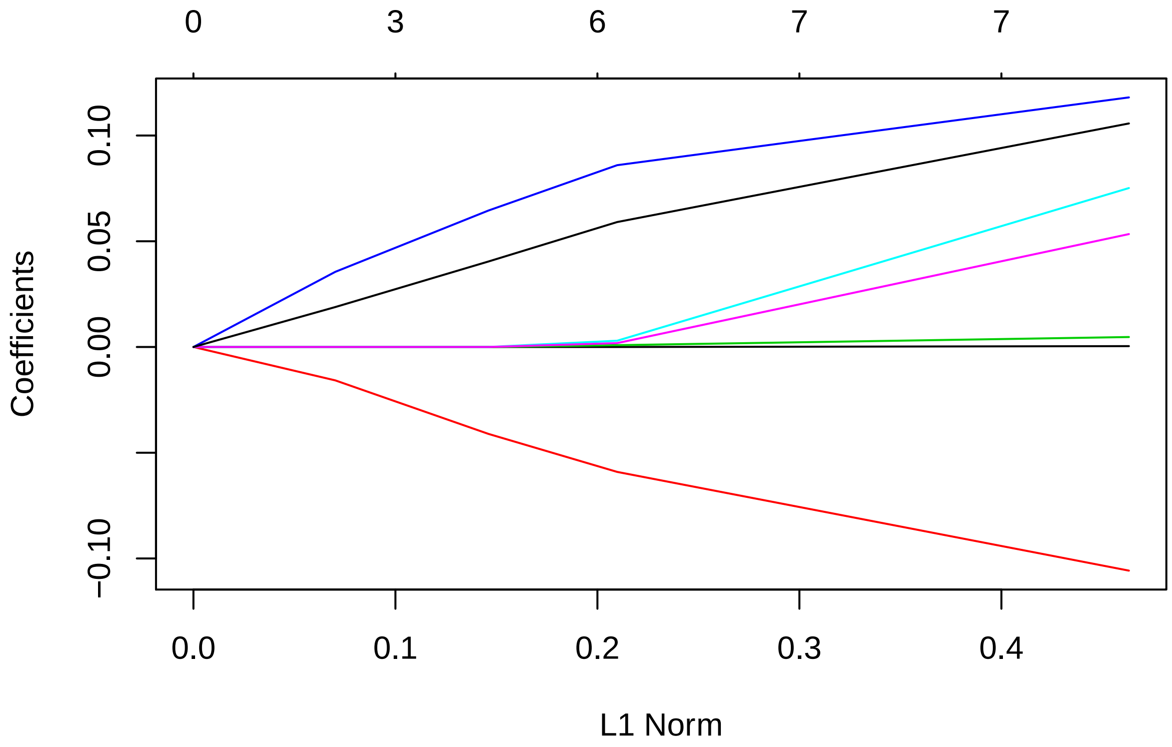
avg_daily_exp_fl_lkbk (MME) has a positive effect on the fall of the patients but it is not statistically significant (p value = 0.1488). The coefficient of gender is negative as female (=1) has a larger fall rate than male (=2) (38% vs 28%). The other factors including age and health probelm indicators also have positive effects on the fall which indicates patients who are older and have health problems have a larger chance to fall.

The signs of the coefficients make sense and agree with intuition.

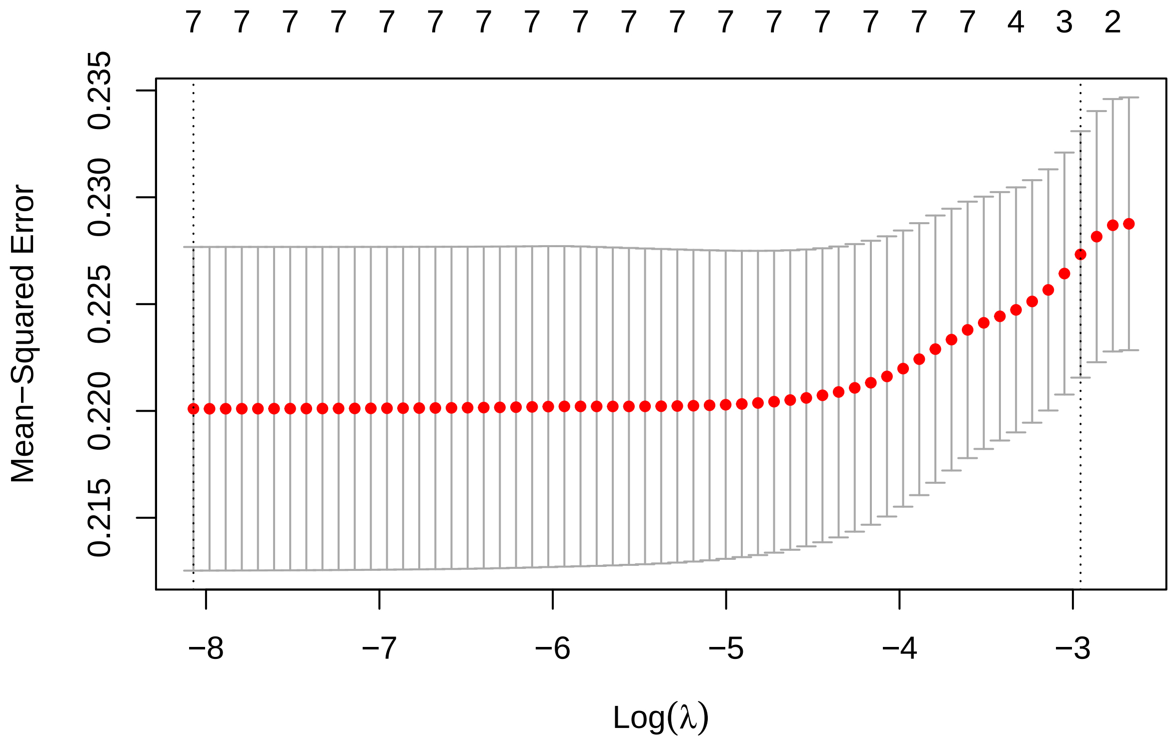
Lasso Regression

```
# x = data.matrix(mmedata.train[,c(1,3:12)])
x = data.matrix(mmedata.train[,c(1,3:4,8,10:12)]) # exclude Patient_Race, Benzo_indicator, CVD_Indicato
y = as.numeric(mmedata.train[,2])-1  # 0 - not fall, 1 - fall
grid =10^ seq (10,-2, length =100)
lasso.mod = glmnet(x, y, alpha = 1, lambda = grid)
plot(lasso.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to unique
## 'x' values
```



```r
set.seed(1)
cv.out = cv.glmnet(x,y,alpha = 1)
plot(cv.out)
```



```
bestlam = cv.out$lambda.min
print(bestlam)
```

```
## [1] 0.0003119257
```

```
# regression coefficients
coef(cv.out, s = "lambda.min")
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                                        1
## (Intercept)                -0.1265494924
## avg_daily_exp_fl_lkbk       0.0005965007
## Patient_Gender             -0.1277890431
## Patient_Age_at_Visit_Date   0.0065259200
## COPD_Indicator              0.1331056651
## Sleep_Apnea_Indicator       0.1092184422
## Cancer_Indicator            0.0777273695
## Psychiatric_Indicator       0.1276829084
```

```
xx = data.matrix(mmedata.test[,c(1,3:4,8,10:12)])
lasso.probs = predict(lasso.mod, s = bestlam, newx= xx)
lasso.pred=rep(0,length(lasso.probs))
lasso.pred[lasso.probs>0.4]=1
```

```
# Confusion Matrix
confusion_mat <- table(mmedata.test$STEADI_q_yn, lasso.pred)
colnames(confusion_mat) <- c("Predicted.Not_Fall", "Predicted.Fall")
rownames(confusion_mat) <- c("Actual.Not_Fall", "Actual.Fall")
print(confusion_mat)
```

```
##                   lasso.pred
##                    Predicted.Not_Fall Predicted.Fall
##   Actual.Not_Fall                 77             19
##   Actual.Fall                     31             21
```

```
TN = confusion_mat[1,1]
FP = confusion_mat[1,2]
FN = confusion_mat[2,1]
TP = confusion_mat[2,2]

paste('Overall accuracy =',round((TN+TP)/(TN+FP+FN+TP),3))
```

```
## [1] "Overall accuracy = 0.662"
```

```
paste('Sensitivity =',round(TP/(TP+FN),3))
```

```
## [1] "Sensitivity = 0.404"
```

```
paste('Specificity =',round(TN/(TN+FP),3))
```

```
## [1] "Specificity = 0.802"
```

The performance of the Lasso model is similar with the logistic regression model.

avg_daily_exp_fl_lkbk has a positive effect on the patient fall.The sign of the coefficents in the Lasso model are consistent with the logistic regression model.

Part 3

```
# Seperate the data based on a threshold of MME and compare "Fall" and "Not Fall" in these two groups

# Contingency table
threshold = quantile(mmedata$avg_daily_exp_fl_lkbk, 0.8) # increase from 0.5 to 0.9
flag.threshold = mmedata$avg_daily_exp_fl_lkbk>threshold

below.threshold = table(mmedata$STEADI_q_yn[!flag.threshold])
above.threshold = table(mmedata$STEADI_q_yn[flag.threshold])

contingency.table = data.frame(below=as.matrix(below.threshold),above=as.matrix(above.threshold))
rownames(contingency.table) = c("Not_fall","Fall")
contingency.table
```

```
##          below above
## Not_fall   315    68
## Fall       157    50
```

```r
# Chi-Square Test of Independence
chisq <- chisq.test(contingency.table)
chisq
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  contingency.table
## X-squared = 3.0516, df = 1, p-value = 0.08066
```

```r
# Fisher's exact test
fisher <- fisher.test(contingency.table)
fisher
```

```
## 
##  Fisher's Exact Test for Count Data
## 
## data:  contingency.table
## p-value = 0.06747
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.9533397 2.2707447
## sample estimates:
## odds ratio 
##   1.474326
```

> With a threshhold of $>$ 80% of MME, the Chi-Square Test and Fisher's exact test show that p-value is less than the significance level of 10%. We can reject the null hypothesis and conclude that there is a significant relationship between the two categorical variables (MME and Fall or not).