BME590FinalProjectEDA

Roujia Wang

2021/04/11

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(tidyverse)
library(ggplot2)
library(lubridate)
library(patchwork)
library(gridExtra)
library(psych)
library(corrplot)
library(ggfortify)
library(factoextra)
library(dplyr) #bind_row
library(class) #knn
library(reshape2) #me
```

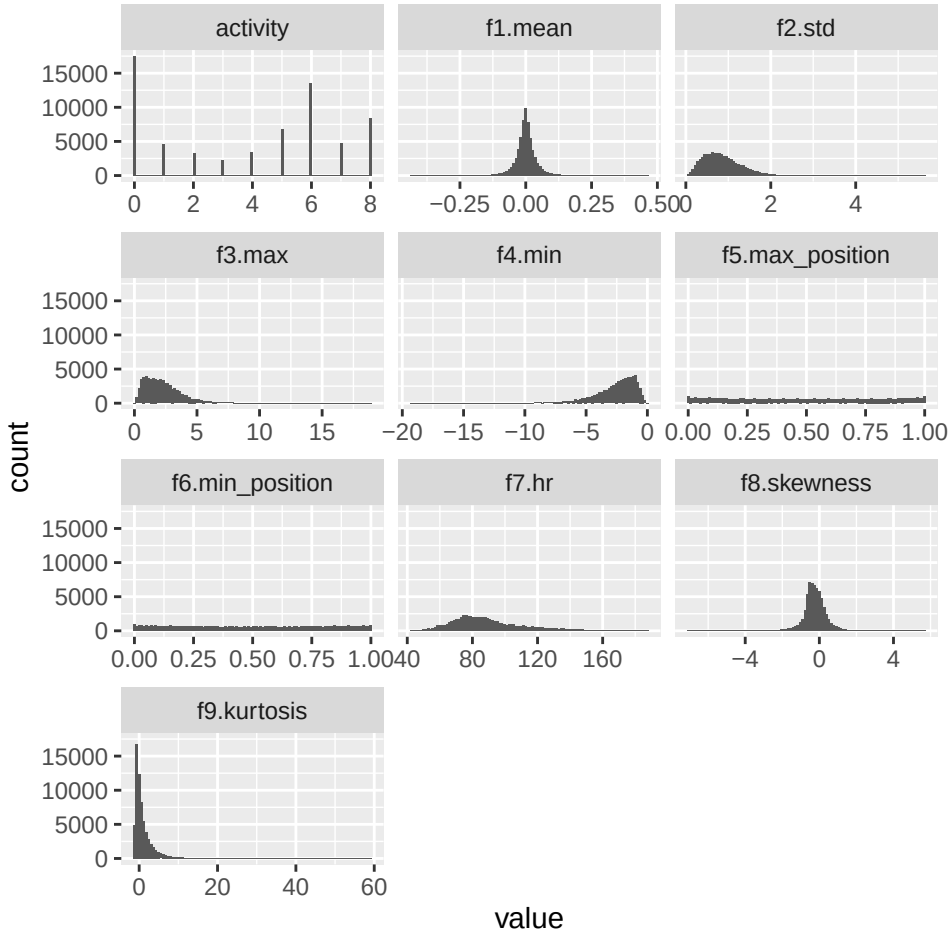
combine all csv files and concatenate into single dataframe

```
# import multiple csv files at once
temp = list.files(pattern="*.csv")
myfiles = lapply(temp, read.csv)
df <- bind_rows(myfiles)
df <- df[, names(df) != "X"]
summary(df)
```

```
##     activity        f1.mean              f2.std            f3.max
##  Min.   :0.000   Min.   :-0.4340166   Min.   :0.02887   Min.   : 0.0695
##  1st Qu.:0.000   1st Qu.:-0.0183473   1st Qu.:0.52229   1st Qu.: 1.1505
##  Median :5.000   Median :-0.0001874   Median :0.79722   Median : 2.0233
##  Mean   :3.837   Mean   : 0.0000072   Mean   :0.87332   Mean   : 2.3249
##  3rd Qu.:6.000   3rd Qu.: 0.0185599   3rd Qu.:1.13038   3rd Qu.: 3.0739
##  Max.   :8.000   Max.   : 0.4626834   Max.   :5.58749   Max.   :18.9332
##      f4.min          f5.max_position  f6.min_position      f7.hr
##  Min.   :-19.34485   Min.   :0.0000   Min.   :0.0000   Min.   : 41.82
```

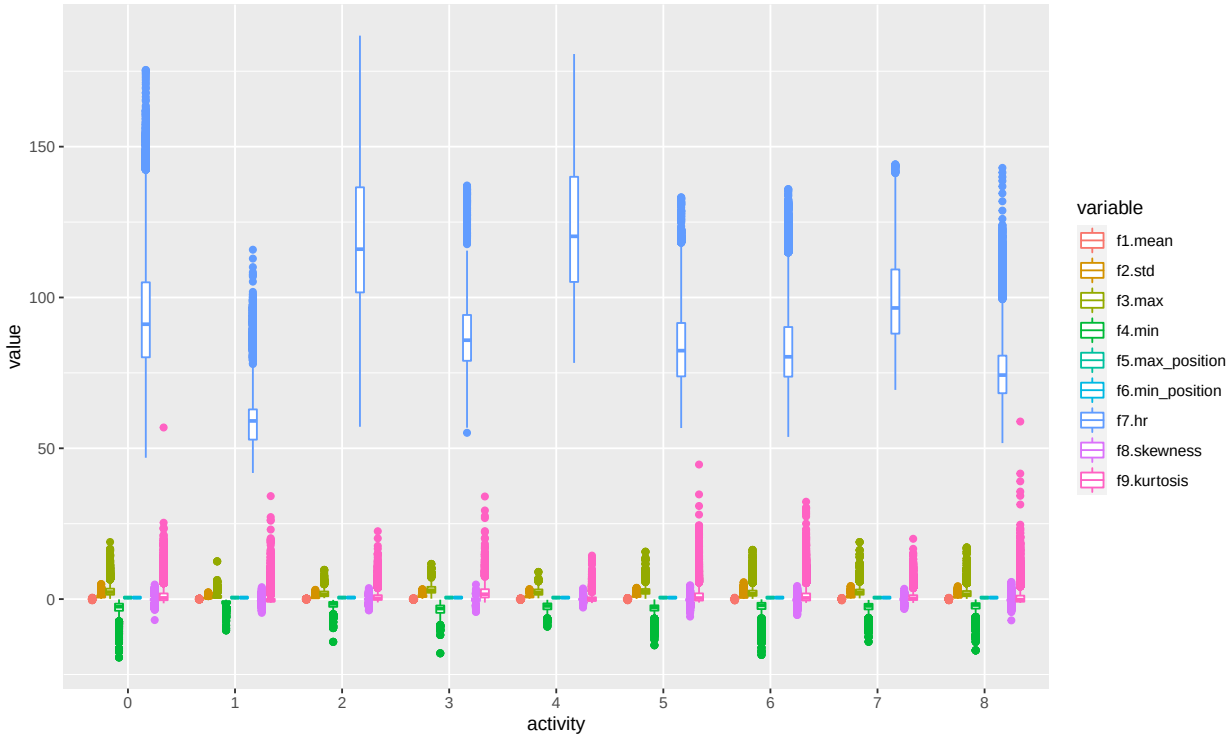
```
##  1st Qu.: -3.44921   1st Qu.:0.2227   1st Qu.:0.2286   1st Qu.: 73.91
##  Median : -2.21189   Median :0.4904   Median :0.4923   Median : 85.22
##  Mean   : -2.64047   Mean   :0.4937   Mean   :0.4951   Mean   : 89.43
##  3rd Qu.: -1.33491   3rd Qu.:0.7638   3rd Qu.:0.7618   3rd Qu.:101.26
##  Max.   : -0.07904   Max.   :0.9982   Max.   :0.9982   Max.   :186.81
##   f8.skewness        f9.kurtosis
##  Min.   :-7.05426   Min.   :-1.3921
##  1st Qu.:-0.55758   1st Qu.:-0.5105
##  Median :-0.26019   Median : 0.2037
##  Mean   :-0.23764   Mean   : 1.0517
##  3rd Qu.: 0.08239   3rd Qu.: 1.6217
##  Max.   : 5.68466   Max.   :58.8658
```

```
temp_df <- gather(df, key="key", value="value")
ggplot(gather(df), aes(value)) +
    geom_histogram(bins = 100) +
    facet_wrap(~key, scales = 'free_x', ncol=3)
```



plot each variables again labels

```
df$activity <- as.factor(df$activity)
dat.m <- melt(df,id.vars='activity', measure.vars=c('f1.mean', 'f2.std',
                                                    'f3.max', 'f4.min',
                                                    'f5.max_position',
                                                    'f6.min_position',
                                                    'f7.hr',
                                                    'f8.skewness',
                                                    'f9.kurtosis'))
ggplot(dat.m) + geom_boxplot(aes(x=activity, y=value, color=variable))
```

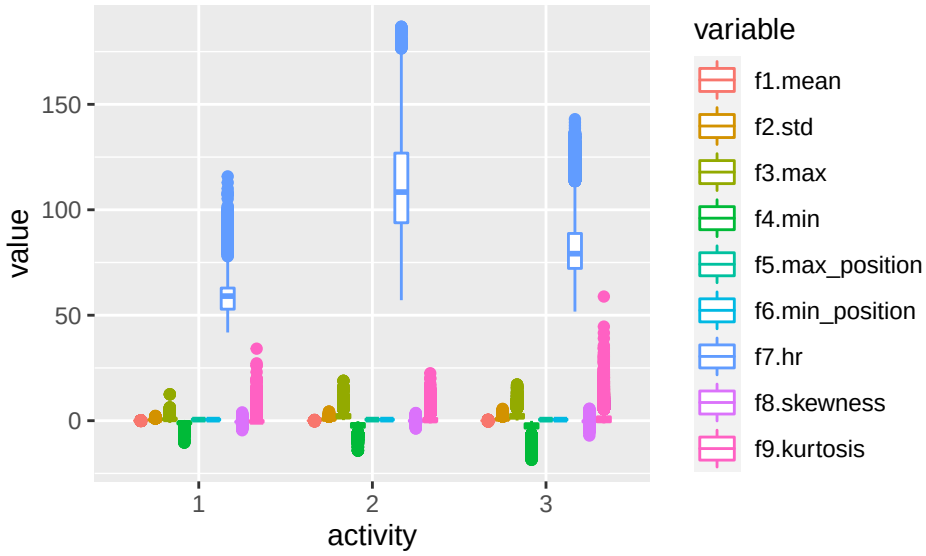


reassign labels to each activity group

```
df$activity <- as.factor(df$activity)
df <- subset(df, df$activity == 1 | df$activity == 2 | df$activity == 3 |
               df$activity ==4 | df$activity == 5| df$activity == 6 | df$activity == 7 |
               df$activity ==8)
# df$Activity
# low = 1
# medium = 3
df$activity[df$activity== 3 | df$activity== 5 | df$activity == 6 | df$activity == 8] <- 3
# high = 2
df$activity[df$activity==2 | df$activity== 4 | df$activity == 7] <- 2
df$activity <- factor(df$activity)
```

plot each variables again labels

```
dat.m <- melt(df,id.vars='activity', measure.vars=c('f1.mean', 'f2.std',
                                                    'f3.max', 'f4.min',
                                                    'f5.max_position',
                                                    'f6.min_position',
                                                    'f7.hr',
                                                    'f8.skewness',
                                                    'f9.kurtosis'))
ggplot(dat.m) + geom_boxplot(aes(x=activity, y=value, color=variable))
```



normalize the variables

```
df_norm <- scale(df[,2:10])
summary(df_norm)
```

```
##     f1.mean             f2.std            f3.max            f4.min
##  Min.   :-9.948958   Min.   :-1.7334   Min.   :-1.3425   Min.   :-8.9044
##  1st Qu.:-0.408482   1st Qu.:-0.7242   1st Qu.:-0.7149   1st Qu.:-0.4183
##  Median :-0.004398   Median :-0.1574   Median :-0.1947   Median : 0.2357
##  Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.416255   3rd Qu.: 0.5242   3rd Qu.: 0.4380   3rd Qu.: 0.6998
##  Max.   :10.603682   Max.   :10.1100   Max.   :10.4436   Max.   : 1.3680
##  f5.max_position     f6.min_position       f7.hr          f8.skewness
##  Min.   :-1.630200   Min.   :-1.65012   Min.   :-1.9819   Min.   :-10.74991
##  1st Qu.:-0.896191   1st Qu.:-0.88937   1st Qu.:-0.6638   1st Qu.: -0.49890
##  Median :-0.007375   Median :-0.01153   Median :-0.2080   Median : -0.05107
##  Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.0000   Mean   :  0.00000
##  3rd Qu.: 0.887327   3rd Qu.: 0.88532   3rd Qu.: 0.4856   3rd Qu.:  0.49562
##  Max.   : 1.660185   Max.   : 1.67223   Max.   : 4.3074   Max.   :  9.38694
##   f9.kurtosis
```

```
##  Min.   :-0.8820
##  1st Qu.:-0.5942
##  Median :-0.3280
##  Mean   : 0.0000
##  3rd Qu.: 0.2053
##  Max.   :22.2154
```

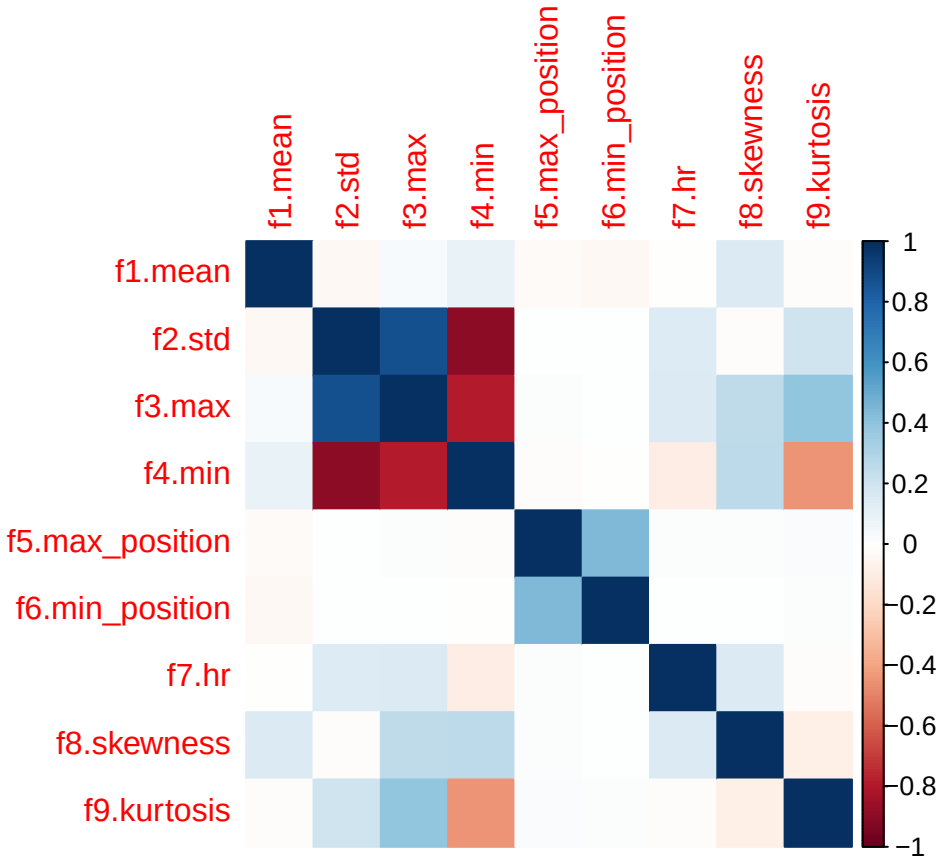
Correlation

1. Correlation using normalized data.

```
M<-cor(df_norm)
head(round(M,2))
```

```
##                 f1.mean f2.std f3.max f4.min f5.max_position f6.min_position
## f1.mean            1.00  -0.04   0.04   0.09           -0.03           -0.03
## f2.std            -0.04   1.00   0.88  -0.89            0.01            0.00
## f3.max             0.04   0.88   1.00  -0.79            0.01            0.01
## f4.min             0.09  -0.89  -0.79   1.00           -0.01           -0.01
## f5.max_position   -0.03   0.01   0.01  -0.01            1.00            0.44
## f6.min_position   -0.03   0.00   0.01  -0.01            0.44            1.00
##                 f7.hr f8.skewness f9.kurtosis
## f1.mean          0.00        0.15       -0.02
## f2.std           0.15       -0.02        0.20
## f3.max           0.15        0.25        0.39
## f4.min          -0.10        0.27       -0.45
## f5.max_position  0.01        0.01        0.02
## f6.min_position  0.01        0.00        0.02
```

```
corrplot(M, method="color")
```

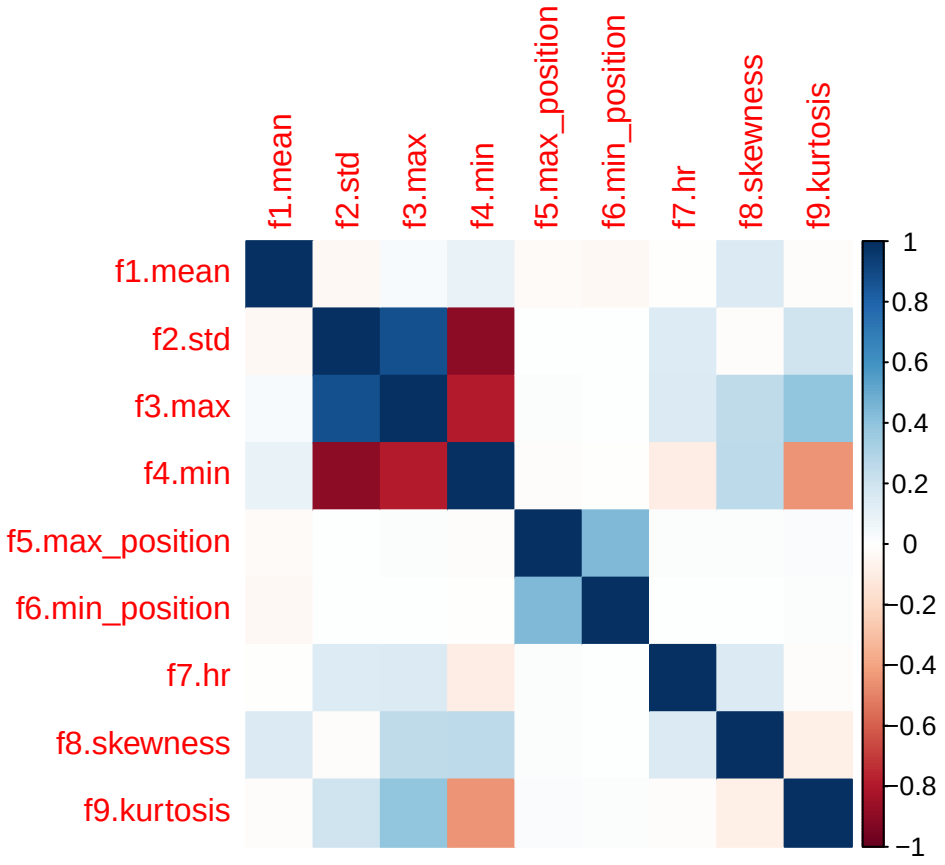


2. Correlation using raw data.

```
M2<-cor(df[,2:10])
head(round(M2,2))
```

```
##                 f1.mean f2.std f3.max f4.min f5.max_position f6.min_position
## f1.mean            1.00  -0.04   0.04   0.09           -0.03           -0.03
## f2.std            -0.04   1.00   0.88  -0.89            0.01            0.00
## f3.max             0.04   0.88   1.00  -0.79            0.01            0.01
## f4.min             0.09  -0.89  -0.79   1.00           -0.01           -0.01
## f5.max_position   -0.03   0.01   0.01  -0.01            1.00            0.44
## f6.min_position   -0.03   0.00   0.01  -0.01            0.44            1.00
##                 f7.hr f8.skewness f9.kurtosis
## f1.mean          0.00        0.15       -0.02
## f2.std           0.15       -0.02        0.20
## f3.max           0.15        0.25        0.39
## f4.min          -0.10        0.27       -0.45
## f5.max_position  0.01        0.01        0.02
## f6.min_position  0.01        0.00        0.02
```

```
corrplot(M2, method="color")
```

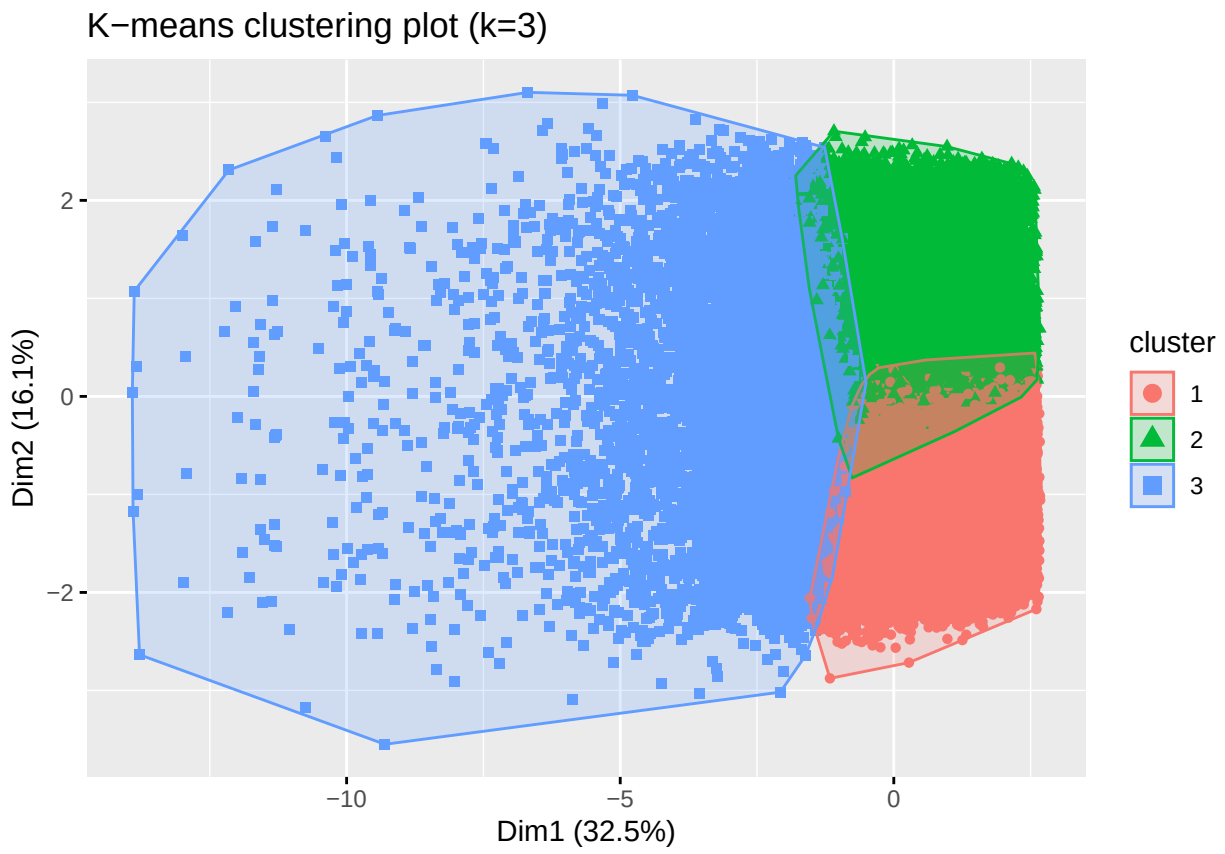


K-Means Clustering

```
df_kmeans <- kmeans(df_norm, centers = 3)
table(df_kmeans$cluster, df$activity)
```

```
##
##          1     2     3
##    1  2385  4255 12077
##    2  2016  4605 11453
##    3   168  2549  7674
```

```
fviz_cluster(df_kmeans, data = df_norm, geom = 'point') +
  ggtitle("K-means clustering plot (k=3)")
```



Feature selection with normalized data

1. Lasso regression

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

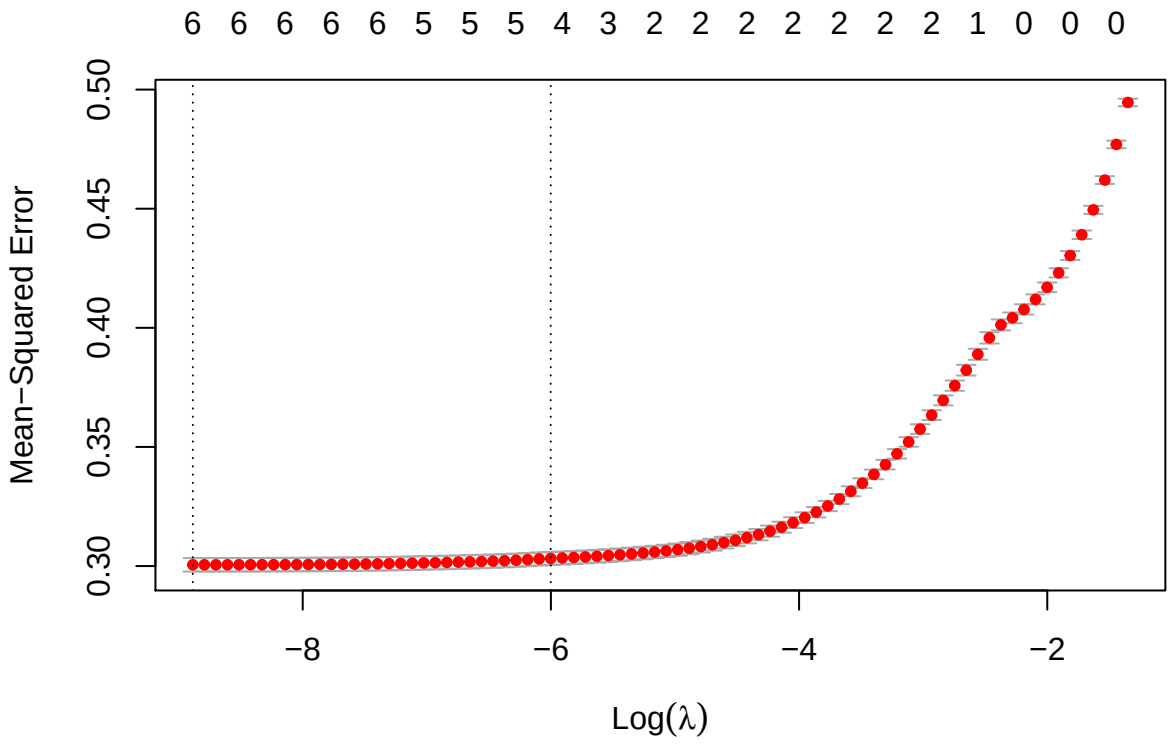
```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-1
```

```
x <- as.matrix(df_norm) # all X vars
y <- as.double(as.matrix(df$activity)) # Only Class

# Fit the LASSO model (Lasso: Alpha = 1)
set.seed(100)
cv.lasso <- cv.glmnet(x, y, family='multinomial', alpha=1, standardize=TRUE, type.measure='mse')
```

```
# Results
plot(cv.lasso)
```




```
# plot(cv.lasso$glmnet.fit, xvar="lambda", label=TRUE)
cat('Min Lambda: ', cv.lasso$lambda.min, '\n 1Sd Lambda: ', cv.lasso$lambda.1se)
```

```
## Min Lambda:  0.0001383767
##  1Sd Lambda:  0.002475074
```

```
df_coef <- coef(cv.lasso, s=cv.lasso$lambda.min)

# See all contributing variables
print(df_coef)
```

```
## $`1`
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)     -2.675710206
## f1.mean          .
## f2.std          -0.601583892
## f3.max           0.748983554
## f4.min           0.638225090
## f5.max_position  0.062357126
## f6.min_position -0.008896926
```

```
## f7.hr           -3.251222667
## f8.skewness     -0.387626596
## f9.kurtosis      .
##
## $`2`
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)      0.528064110
## f1.mean         -0.007246876
## f2.std           .
## f3.max           .
## f4.min          -0.582910153
## f5.max_position -0.006262255
## f6.min_position  .
## f7.hr            1.791311762
## f8.skewness      0.442039063
## f9.kurtosis     -0.230933002
##
## $`3`
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)      2.1476460963
## f1.mean          0.0321094457
## f2.std           0.7001003209
## f3.max          -0.0500928157
## f4.min           .
## f5.max_position  .
## f6.min_position  0.0002015384
## f7.hr            .
## f8.skewness      .
## f9.kurtosis      0.1188822787
```

2. Forward and backward stepwise selection

```
trainData <- df
trainData$activity <- as.double(trainData$activity)
trainData[, 2:10] <- df_norm

# Step 1: Define base intercept only model
base.mod <- lm(activity ~ 1 , data=trainData)

# Step 2: Full model with all predictors
all.mod <- lm(activity ~ . , data= trainData)

# Step 3: Perform step-wise algorithm. direction='both' implies both forward and backward stepwise
stepMod <- step(base.mod, scope = list(lower = base.mod, upper = all.mod),
                direction = "both", trace = 0, steps = 1000)

# Step 4: Get the shortlisted variable.
shortlistedVars <- names(unlist(stepMod[[1]]))
shortlistedVars <- shortlistedVars[!shortlistedVars %in% "(Intercept)"] # remove intercept

# Show
print(shortlistedVars)
```

```
## [1] "f2.std"          "f7.hr"           "f9.kurtosis"     "f8.skewness"
## [5] "f3.max"          "f4.min"          "f1.mean"         "f5.max_position"
## [9] "f6.min_position"
```

Feature selection with raw data

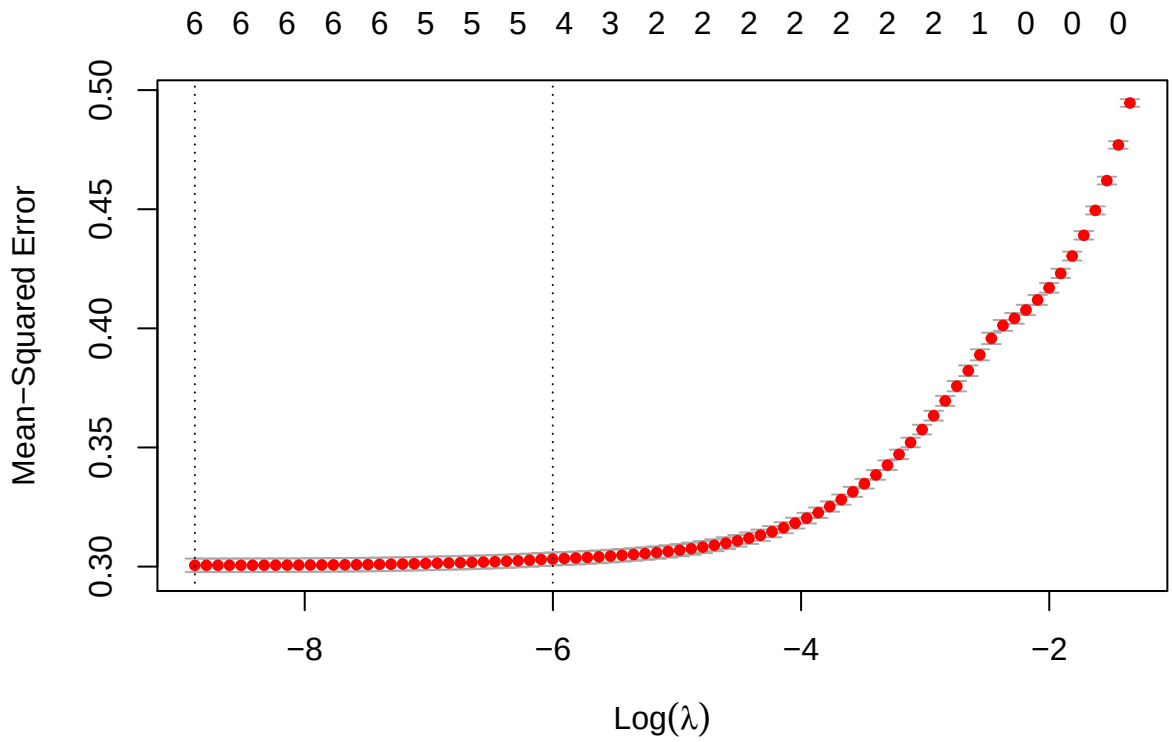
1. Lasso regression

```
library(glmnet)

x <- as.matrix(df[, 2:10]) # all X vars
y <- as.double(as.matrix(df$activity)) # Only Class

# Fit the LASSO model (Lasso: Alpha = 1)
set.seed(100)
cv.lasso_raw <- cv.glmnet(x, y, family='multinomial', alpha=1, standardize=TRUE, type.measure='mse')
```

```
# Results
plot(cv.lasso_raw)
```




```
# plot(cv.lasso$glmnet.fit, xvar="lambda", label=TRUE)
cat('Min Lambda: ', cv.lasso_raw$lambda.min, '\n 1Sd Lambda: ', cv.lasso_raw$lambda.1se)
```

```
## Min Lambda:  0.0001383767
##  1Sd Lambda:  0.002475074
```

```r
df_coef_raw <- coef(cv.lasso_raw, s=cv.lasso_raw$lambda.min)
# See all contributing variables
print(df_coef_raw)
```

```
## $`1`
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)      8.87893363
## f1.mean          .
## f2.std          -1.28175100
## f3.max           0.46886248
## f4.min           0.35578691
## f5.max_position  0.20554598
## f6.min_position -0.02961154
## f7.hr           -0.14103071
## f8.skewness     -0.61273518
## f9.kurtosis      .
##
## $`2`
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)     -8.30676797
## f1.mean         -0.16610065
## f2.std           .
## f3.max           .
## f4.min          -0.32495087
## f5.max_position -0.02064209
## f6.min_position  .
## f7.hr            0.07770307
## f8.skewness      0.69874690
## f9.kurtosis     -0.08863993
##
## $`3`
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)     -0.5721656638
## f1.mean          0.7359584494
## f2.std           1.4916527784
## f3.max          -0.0313580209
## f4.min           .
## f5.max_position  .
## f6.min_position  0.0006707782
## f7.hr            .
## f8.skewness      .
## f9.kurtosis      0.0456310584
```

2. Forward and backward stepwise selection

```r
trainData <- df
trainData$activity <- as.double(trainData$activity)

# Step 1: Define base intercept only model
base.mod <- lm(activity ~ 1 , data=trainData)
```

```r
# Step 2: Full model with all predictors
all.mod <- lm(activity ~ . , data= trainData)

# Step 3: Perform step-wise algorithm. direction='both' implies both forward and backward stepwise
stepMod <- step(base.mod, scope = list(lower = base.mod, upper = all.mod),
                direction = "both", trace = 0, steps = 1000)

# Step 4: Get the shortlisted variable.
shortlistedVars <- names(unlist(stepMod[[1]]))
shortlistedVars <- shortlistedVars[!shortlistedVars %in% "(Intercept)"] # remove intercept

# Show
print(shortlistedVars)
```

```
## [1] "f2.std"          "f7.hr"           "f9.kurtosis"     "f8.skewness"
## [5] "f3.max"          "f4.min"          "f1.mean"         "f5.max_position"
## [9] "f6.min_position"
```