

```

# Loading libraries
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.2

library(caret)

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 3.6.2

```

Preparing dataset

```

# Preparing data
# From the root folder
root_folder = "Datasets_Healthy_Older_People"

# Read all available directories
data_dirs = list.files(path=root_folder)

df_all <- data.frame()

# Looping through all files and reading file by file
for(folder in data_dirs) {
  folder_path = file.path(root_folder, folder)
  data_files = list.files(path=folder_path)
  for (f in data_files) {
    if(f != "README.txt") {
      m_f = substr(f, nchar(f), nchar(f))
      df = read.csv(file.path(folder_path, f), header = FALSE)
      df$Sex = m_f
      df_all = rbind(df_all, df)
    }
  }
}

col_names <- c("Time", "AccFrontal", "AccVertical", "AccLateral",
             "Id", "RSSI", "Phase", "Frequency", "Activity", "Sex")
names(df_all) <- col_names

head(df_all)

```

```

##   Time AccFrontal AccVertical AccLateral Id  RSSI  Phase Frequency Activity Sex
## 1 0.00    0.27203     1.00820  -0.082102  1 -63.5  2.4252    924.25      1   M
## 2 0.50    0.27203     1.00820  -0.082102  1 -63.0  4.7369    921.75      1   M
## 3 1.50    0.44791     0.91636  -0.013684  1 -63.5  3.0311    923.75      1   M
## 4 1.75    0.44791     0.91636  -0.013684  1 -63.0  2.0371    921.25      1   M
## 5 2.50    0.34238     0.96229  -0.059296  1 -63.5  5.8920    920.25      1   M
## 6 3.25    0.34238     0.96229  -0.059296  4 -56.5  1.2563    924.75      1   M

```

```

tail(df_all)

##           Time AccFrontal AccVertical AccLateral Id  RSSI Phase Frequency
## 75123 442.50     0.90520     0.64076 -0.082102 1 -63.5 6.2571 925.75
## 75124 444.00     0.90520     0.64076 -0.082102 1 -59.5 1.9574 922.25
## 75125 449.00     0.90520     0.64076 -0.082102 1 -59.0 6.2540 922.25
## 75126 451.50     0.42446     0.91636 -0.161920 1 -53.0 5.7156 922.25
## 75127 451.58     0.42446     0.91636 -0.161920 1 -52.0 5.8982 922.25
## 75128 452.25     0.58862     0.87042 -0.082102 1 -60.0 5.7033 925.75
##          Activity Sex
## 75123        2   F
## 75124        2   F
## 75125        2   F
## 75126        4   F
## 75127        4   F
## 75128        4   F

```

```
summary(df_all)
```

```

##           Time          AccFrontal          AccVertical          AccLateral
## Min.    : 0.0      Min.    :-0.7481      Min.    :-0.553490      Min.    :-1.33640
## 1st Qu.: 121.2    1st Qu.: 0.3424      1st Qu.: -0.002297    1st Qu.: -0.18473
## Median  : 250.7    Median  : 0.6824      Median  : 0.215880      Median  : -0.07070
## Mean    : 299.1    Mean    : 0.7142      Mean    : 0.345199      Mean    : -0.21748
## 3rd Qu.: 402.5    3rd Qu.: 1.1045      3rd Qu.: 0.858940      3rd Qu.: 0.03193
## Max.    :1739.4    Max.    : 1.5032      Max.    : 2.030200      Max.    : 1.21780
##          Id          RSSI          Phase          Frequency
## Min.    :1.00      Min.    :-72.00      Min.    :0.000      Min.    :920.2
## 1st Qu.:1.00      1st Qu.: -62.00     1st Qu.: 1.032      1st Qu.: 921.2
## Median  :3.00      Median  : -58.00     Median  : 2.767      Median  : 922.8
## Mean    :2.41      Mean    : -58.28     Mean    : 3.157      Mean    : 922.7
## 3rd Qu.:3.00      3rd Qu.: -56.00     3rd Qu.: 5.359      3rd Qu.: 924.2
## Max.    :4.00      Max.    : -38.50     Max.    : 6.282      Max.    : 925.8
##          Activity       Sex
## Min.    :1.000      Length:75128
## 1st Qu.:2.000      Class :character
## Median  :3.000      Mode  :character
## Mean    :2.528
## 3rd Qu.:3.000
## Max.    :4.000

```

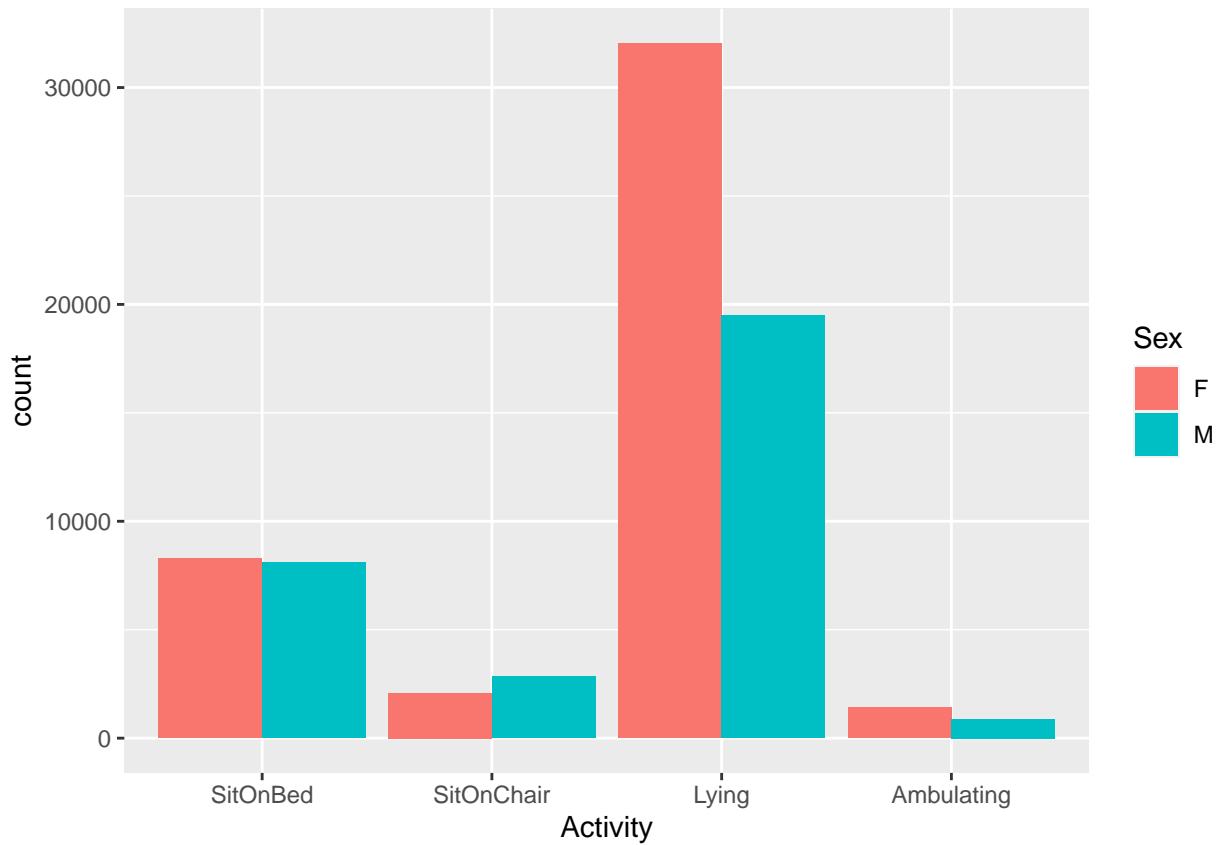
```
df_all$Time <- NULL
```

```

# Printing distribution of various activities

levels = c("SitOnBed", "SitOnChair", "Lying", "Ambulating")
g <- ggplot(df_all, aes(x=Activity, fill=Sex))
g <- g + geom_bar(position = "dodge")
g <- g + scale_x_discrete(limits=c("1", "2", "3", "4"), labels=levels)
g

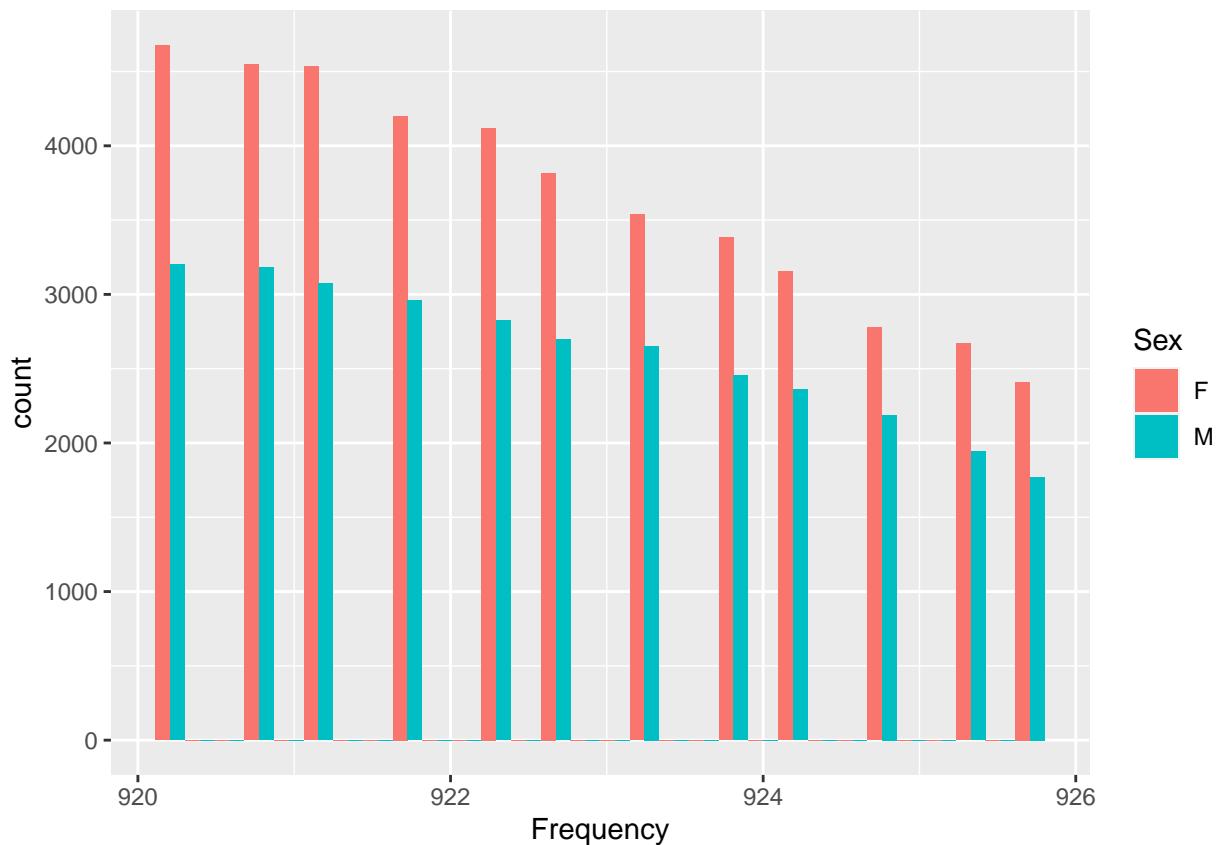
```



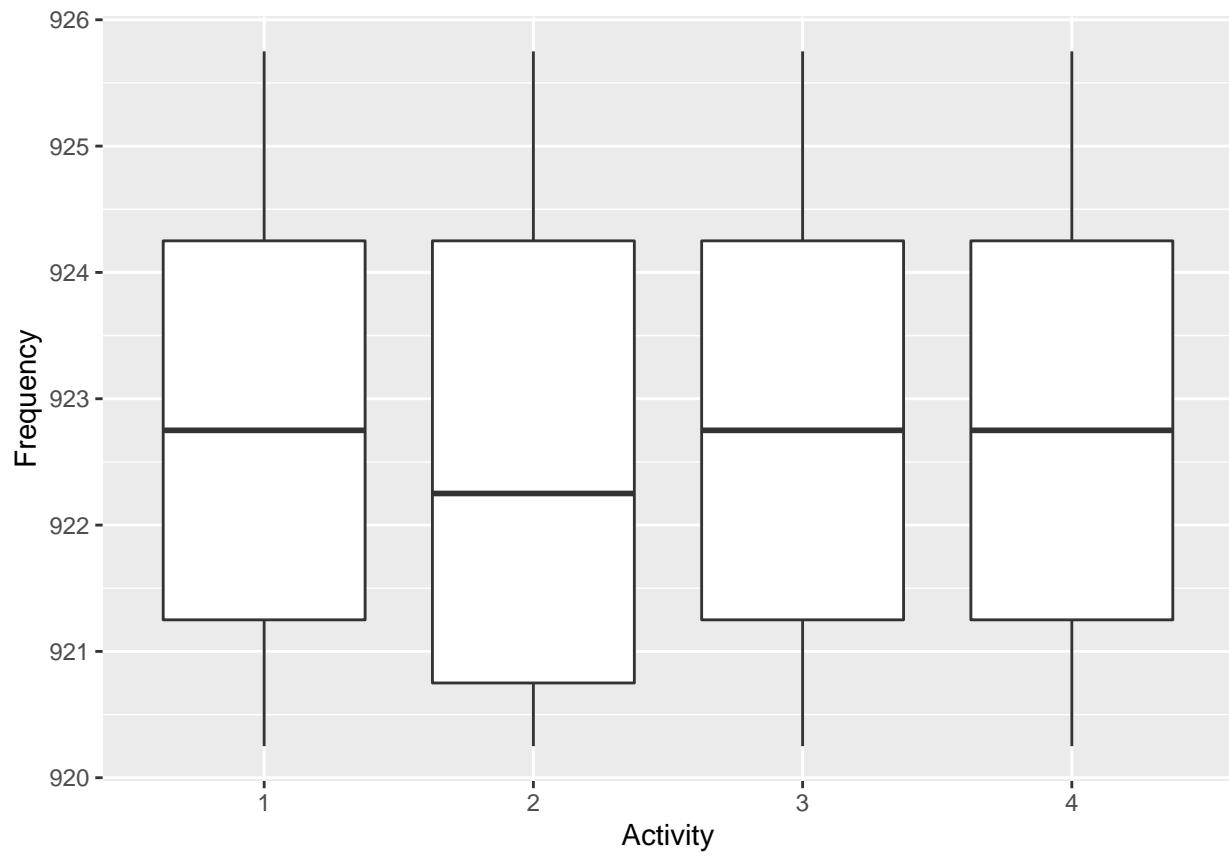
```
# Displaying how Frequency is distributed across Gender
```

```
g <- ggplot(df_all, aes(x=Frequency, fill=Sex))  
g <- g + geom_histogram(position = "dodge")  
g
```

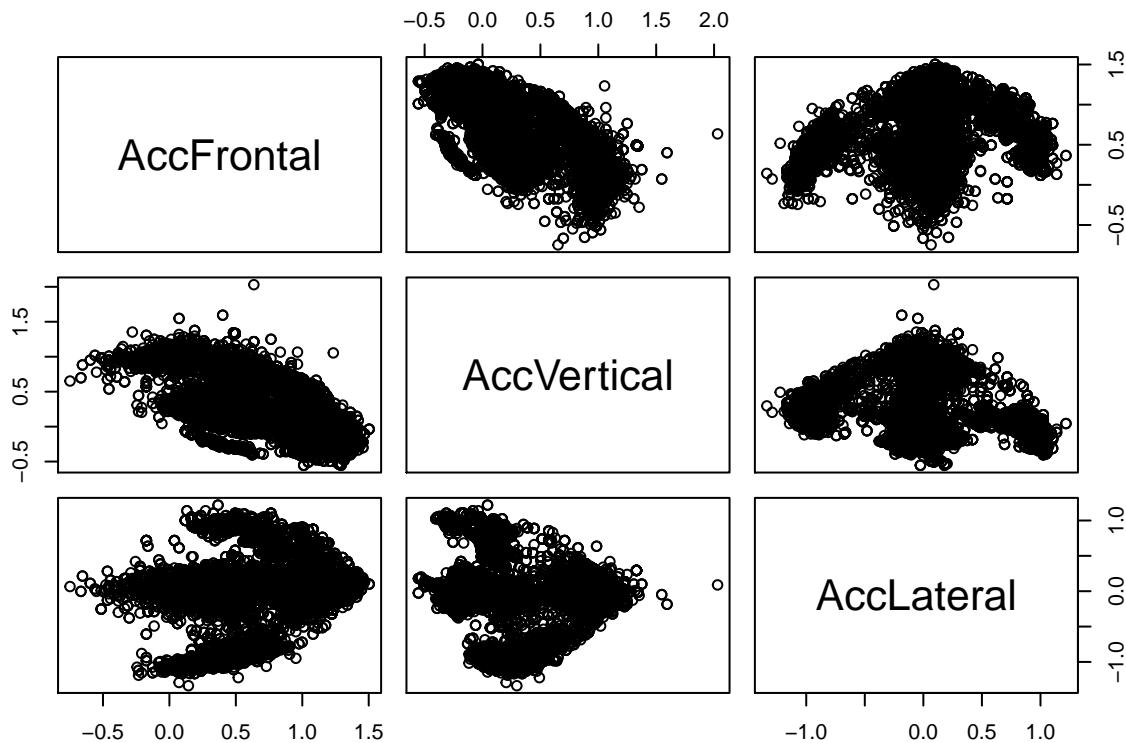
```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



```
# Displaying how Frequency is distributed across various activity
g <- ggplot(df_all, aes(x=as.factor(Activity), y = Frequency))
g <- g + geom_boxplot() + xlab("Activity")
g
```



```
# Displaying relationship across acceleration in various directions
pairs(df_all[, c("AccFrontal", "AccVertical", "AccLateral")])
```



```

# Converting it as categorical variable
df_all$Activity <- factor(df_all$Activity)
y <- df_all$Activity
# Splitting data into train and test set
index <- createDataPartition(y = df_all$Activity, p = 0.70, list = FALSE)
X_train <- df_all[index, ]
X_test <- df_all[-index, ]

y_train <- y[index]
y_test <- y[-index]

# fitting random forest model on the train data set
rf <- train(Activity ~., data = X_train, method = "rpart",
            preProcess = c("center","scale"),
            parms = list(split="information"),
            tuneLength = 10, metric='Accuracy')

# Predicting it on the test set
rf_pred <- predict(rf, newdata = X_test )

# Evaluating model performance
confusionMatrix(rf_pred, X_test$Activity )

## Confusion Matrix and Statistics
##

```

```

##             Reference
## Prediction      1      2      3      4
##               1 4826    145     22   345
##               2    52   1313      0    32
##               3    23       0 15427      1
##               4    20      15      7   309
##
## Overall Statistics
##
##                         Accuracy : 0.9706
##                         95% CI : (0.9683, 0.9728)
## No Information Rate : 0.6858
## P-Value [Acc > NIR] : < 2.2e-16
##
##                         Kappa : 0.938
##
## Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                         Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity              0.9807  0.89138  0.9981  0.4497
## Specificity               0.9709  0.99601  0.9966  0.9980
## Pos Pred Value            0.9041  0.93987  0.9984  0.8803
## Neg Pred Value            0.9945  0.99243  0.9959  0.9829
## Prevalence                 0.2184  0.06536  0.6858  0.0304
## Detection Rate             0.2141  0.05826  0.6845  0.0137
## Detection Prevalence       0.2369  0.06199  0.6856  0.0155
## Balanced Accuracy          0.9758  0.94370  0.9974  0.7239

```

```
# Nearest Shrunken Centroids
# fitting knn model on the train data set
pam <- train(Activity ~., data = X_train, method = "pam",
               preProcess = c("center", "scale"), tuneLength = 10)
```

```
# Predicting it on the test set  
pam_pred <- predict(pam, newdata = X_test )  
  
# Evaluating model performance  
confusionMatrix(pam_pred, X_test$Activity )
```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction    1     2     3     4
##           1 3946   649     2 612
##           2     0     0     0     0
##           3  975   824 15454   75
##           4     0     0     0     0
##
## Overall Statistics

```

```

##
##          Accuracy : 0.8608
##          95% CI : (0.8562, 0.8653)
##  No Information Rate : 0.6858
##  P-Value [Acc > NIR] : < 2.2e-16
##
##          Kappa : 0.6703
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##          Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity      0.8019  0.00000  0.9999  0.00000
## Specificity       0.9283  1.00000  0.7353  1.00000
## Pos Pred Value    0.7575      NaN     0.8919      NaN
## Neg Pred Value    0.9437  0.93464  0.9996  0.96952
## Prevalence        0.2184  0.06536  0.6858  0.03048
## Detection Rate    0.1751  0.00000  0.6857  0.00000
## Detection Prevalence 0.2311  0.00000  0.7689  0.00000
## Balanced Accuracy  0.8651  0.50000  0.8676  0.50000

```

1