

# Analysis of HAR data

This R function analyses the data which were available from the research on human activity recognition (HAR). The function combines training and test set into one whole set, calculates averages, which are based on a subject and its activity, and writes everything in one table named “mean\_by\_subj\_act.txt”. It also calculates averages based on subject (“mean\_by\_subj.txt”) or activity (“mean\_by\_act.txt”) only.

The function assumes that the test and training data can be found in subdirectories test and training, respectively.

```
analyseAll <- function(destsa="mean_by_subj_act.txt", dests="mean_by_subj.txt",
                      desta="mean_by_act.txt") {
```

Read the activity data *a*, subject data *sub* and data set *test*. We use Roger Peng’s classes trick to speed things up with the big data set.

```
  a <- read.table("test/y_test.txt")
  sub <- read.table("test/subject_test.txt")
  test5rows <- read.table("test/X_test.txt", nrows =5)
  classes <- sapply(test5rows, class)
  test <- read.table("test/X_test.txt", colClasses = classes)
```

Now bind subject, activity and data to a nice test set:

```
  atest <- cbind(a,test)
  testset <- cbind(sub,atest)
```

Redo most of that for the training set and then bind the whole set.

```
  a <- read.table("train/y_train.txt")
  sub <- read.table("train/subject_train.txt")
  train <- read.table("train/X_train.txt", colClasses = classes)
  atrain <- cbind(a,train)
  trainset <- cbind(sub,atrain)
  wholeset <- rbind(trainset,testset)
```

Set column names, which come from “features.txt”.

```
  features <- read.table("features.txt")
  feat2 <- as.character(features[,2])
  mycolnames <- c("subject","activity", feat2)
  colnames(wholeset) <- mycolnames
```

Quick and dirty replacement of activity numbers by useful names. The activities are: 1 walk, 2 walk upstairs, 3 walk downstairs, 4 sit, 5 stand, 6 lie.

```
  wholeset$activity<- gsub("1","walk",wholeset$activity)
  wholeset$activity<- gsub("2","walk up",wholeset$activity)
  wholeset$activity<- gsub("3","walk down",wholeset$activity)
  wholeset$activity<- gsub("4","sit",wholeset$activity)
  wholeset$activity<- gsub("5","stand",wholeset$activity)
  wholeset$activity<- gsub("6","lie",wholeset$activity)
```

Tidy up by eliminating columns that don't have *mean()*, *Mean* or *std()* as part of their names. First we must find all desired column names by grepping for *mean()*, *Mean* and *std()*, but we ignore names with *meanFreq* entries.

```
means <- feat2[grepl("mean[~a-zA-Z]",feat2)]
Means <- feat2[grepl("Mean",feat2)]
stds <- feat2[grepl("std",feat2)]
all <- c(means,Means,stds)
names <- c("subject","activity",all)
```

We read all column names and only keep those entries which also show up in the vector *names*:

```
cols <- as.character(colnames(wholeset))
mycols <- cols[cols %in% c(names)]
```

Through the last operation we avoided any reordering of columns, i.e. the standard deviation column of a data set will still directly follow the column of this data set's mean. Now we only choose those columns with mean and std data, but we still get 75 columns (subject, activity + 73 data columns) in total.

```
wholeset2 <- wholeset[, mycols]
```

Time to calculate the averages by subject and activity and put all that in a nice(r) table. We do all columns in one go by having aggregate work on columns 3-75. Then, just for fun, we do the same, this time based on subject or activity only. Finally, we write the tables to three respective destination files.

```
finaltable <- aggregate(wholeset2[,3:75],list(Subject = wholeset2$subject,
      Activity = wholeset2$activity), mean)
tact<-aggregate(wholeset2[,3:75],list(Activity = wholeset2$activity), mean)
tsub <- aggregate(wholeset2[,3:75],list(Subject = wholeset2$subject), mean)
write.table(finaltable, destsa, row.name = FALSE)
write.table(tact, desta, row.name = FALSE)
write.table(tsub, dests, row.name = FALSE)
}
```

And we're done!