

# Code book for the dataset “analysis\_dataset.txt” (assignment 2 of the Course “Getting and Cleaning data”

## Table of content

Table of content .....	1
Original Data Source(s) .....	1
Dataset preparation .....	1
Preparation and cleaning process.....	1
Dataset preparation script .....	2
Description of the dataset.....	3

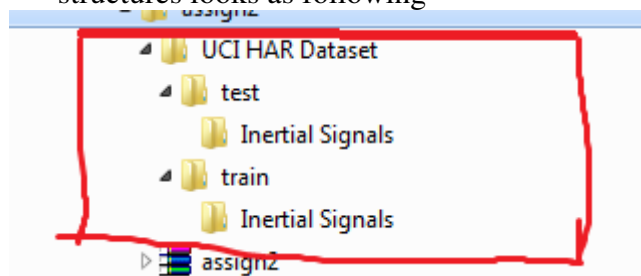
## Original Data Source(s)

Description of the data: the data contain results of the experiments performed on 30 persons. 3-axes accelerations (body and gravitational) and 3-axes rotational velocities were measured and recorded in a form of 2.56s time-series (128 instances) for each measurement point. The time series were subjected to the fast Fourier transformation (FFT) and resulting factors (561 per data point) were stored in the original data set. The dataset was split into the training and test ones and stored in the files, which were used in the present exercise. All the details of the original dataset can be found in following website: <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones#>

## Dataset preparation

### Preparation and cleaning process

1. Downloading the original dataset and unpacking the files. The resulting folder structures looks as following



2. Reading the training and test datasets **X\_test.txt** and **X\_train.txt** files and the corresponding **y\_test.txt** and **y\_train.txt** files containing the activity labels and the **subject\_train.txt** and **subject\_test.txt** files containing the subject identification data.
3. Reading the activity identification table **activity\_labels.txt** and names of the variables **features.txt**.

4. X..., y..., subject... and subject... train and test data frames are merged, numerical activities ids are substituted with their verbal equivalents, then the subject and labels variables were attached to the FFT variables data frame from the left.
5. Names containing “mean” and “std” character sequences are identified and corresponding subset of the merged data frame is made. Corresponding (selected) names are assigned to the data frame columns. Characters sequence “()” is removed from the names.
6. Averages over the activities and subjects are generated and resulting dataset is written into the file “analysis\_dataset.txt” with the following structure (subjects, activity labels and 79 data columns):

subjects	labels	tBodyAcc-mean-X	tBodyAcc-mean-Y	tBodyAcc-mean-Z ....
1	LAYING	0.22159824394	-0.0405139534294	-0.11320355358 .....
1	SITTING	0.261237565425532	-0.00130828765170213	-0.104544182255319 .....
1	STANDING	0.278917629056604	-0.0161375901037736	-0.110601817735849 .....
1	WALKING	0.277330758736842	-0.0173838185273684	-0.111148103547368 .....

Description of each variable is given further in the present document.

## Dataset preparation script

```
#This script fulfills (hopefully) requirements of Assignment 2 for Data Science Course 3
##Downloading and unpacking the data
siteUrl="https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"
if(!file.exists("assign2.zip"))download.file(siteUrl,destfile = "assign2.zip", mode="wb")
if(!dir.exists("UCI HAR Dataset")){
  unzip("assign2.zip", files = NULL, list = FALSE, overwrite = TRUE,
    junkpaths = FALSE, exdir = ".", unzip = "internal",
    setTimes = FALSE)}

##Reading training dataset
dir_tr="UCI HAR Dataset\\train\\"
dat_train<-read.table(paste(dir_tr, "X_train.txt",sep=""))
labels_train<-read.table(paste(dir_tr, "y_train.txt",sep=""),colClasses = "factor")
subject_train<-read.table(paste(dir_tr, "subject_train.txt",sep=""))

##Reading test dataset
dir_test="UCI HAR Dataset\\test\\"
dat_test<-read.table(paste(dir_test, "X_test.txt",sep=""))
labels_test<-read.table(paste(dir_test, "y_test.txt",sep=""),colClasses = "factor")
subject_test<-read.table(paste(dir_test, "subject_test.txt",sep=""))

##reading common for both datasets info
features<-read.table(file="UCI HAR Dataset\\features.txt",sep="")
activity_labels<-read.table(file="UCI HAR Dataset\\activity_labels.txt",sep="")

##merging datasets
dat<-rbind(dat_train, dat_test)
labels<-rbind(labels_train,labels_test)
subjects<-rbind(subject_train,subject_test)
```

```

##Subsetting dataset by extracting columns with means and stds only
needed_cols<-grepl("mean", features[,2]) | grepl("std", features[,2])
dat<-dat[,needed_cols]

##Tiding up and attaching activity names, labels and subjects
for(i in 1:6){
  labels[,1]<-sub(activity_labels[i,1],activity_labels[i,2], labels[,1])
}
dat<-cbind(subjects[,1],labels[,1], dat)
names_inter<-gsub("()", "", features[,2],fixed=TRUE)
names(dat)<-c("subjects", "labels", as.character(names_inter[needed_cols]))

##THIS function generates averages over the Activity for a given dataframe column
fun<-function(dat, subj , j ){
  tapply(as.vector(dat[dat$subjects==subj, j]),
        dat$labels[dat$subjects==subj], mean)
}

##generating a resulting dataset of averages for Activities and Subjects
a<-3:dim(dat)[2] ## columns to average
dat_result<-data.frame()
for (j in 1:max(dat$subjects)){ ## looping over subjects
  interim<-sapply(X=a,FUN=fun, dat=dat, subj=j, simplify=TRUE)
  xx<-cbind(rep(j,6), row.names(interim),interim)
  dat_result<-rbind(dat_result,xx)
}
names(dat_result)<-names(dat)

##if(!file.exists("analysis_dataset.txt"))
  write.table(dat_result, file="analysis_dataset.txt", row.names=FALSE,quote = FALSE)

##control: ww<-read.table(file="assign2.txt", colClasses = "character")

```

## ***Description of the dataset***

The dataset consists of the following variables (quotes omitted, all lowercase, the text after the # sign is a description and does not belong to the variable name):

```

--subjects # number representing the personal identifier of the test participant (1-30)
--labels # the name of the activity, the measurements has been taken for (there are six of
them : (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING,
STANDING, LAYING)
--the following 79 variables are representing the mean and average variable corresponding
to the FFT of the original time series for nine degrees of freedom as described above. The
names are self-explanatory. Detail description of their meaning is beyond the scope of the
exercise):
tBodyAcc-mean-X
tBodyAcc-mean-Y
tBodyAcc-mean-Z
tBodyAcc-std-X

```

tBodyAcc-std-Y  
tBodyAcc-std-Z  
tGravityAcc-mean-X  
tGravityAcc-mean-Y  
tGravityAcc-mean-Z  
tGravityAcc-std-X  
tGravityAcc-std-Y  
tGravityAcc-std-Z  
tBodyAccJerk-mean-X  
tBodyAccJerk-mean-Y  
tBodyAccJerk-mean-Z  
tBodyAccJerk-std-X  
tBodyAccJerk-std-Y  
tBodyAccJerk-std-Z  
tBodyGyro-mean-X  
tBodyGyro-mean-Y  
tBodyGyro-mean-Z  
tBodyGyro-std-X  
tBodyGyro-std-Y  
tBodyGyro-std-Z  
tBodyGyroJerk-mean-X  
tBodyGyroJerk-mean-Y  
tBodyGyroJerk-mean-Z  
tBodyGyroJerk-std-X  
tBodyGyroJerk-std-Y  
tBodyGyroJerk-std-Z  
tBodyAccMag-mean  
tBodyAccMag-std  
tGravityAccMag-mean  
tGravityAccMag-std  
tBodyAccJerkMag-mean  
tBodyAccJerkMag-std  
tBodyGyroMag-mean  
tBodyGyroMag-std  
tBodyGyroJerkMag-mean  
tBodyGyroJerkMag-std  
fBodyAcc-mean-X  
fBodyAcc-mean-Y  
fBodyAcc-mean-Z  
fBodyAcc-std-X  
fBodyAcc-std-Y  
fBodyAcc-std-Z  
fBodyAcc-meanFreq-X  
fBodyAcc-meanFreq-Y  
fBodyAcc-meanFreq-Z  
fBodyAccJerk-mean-X

fBodyAccJerk-mean-Y  
fBodyAccJerk-mean-Z  
fBodyAccJerk-std-X  
fBodyAccJerk-std-Y  
fBodyAccJerk-std-Z  
fBodyAccJerk-meanFreq-X  
fBodyAccJerk-meanFreq-Y  
fBodyAccJerk-meanFreq-Z  
fBodyGyro-mean-X  
fBodyGyro-mean-Y  
fBodyGyro-mean-Z  
fBodyGyro-std-X  
fBodyGyro-std-Y  
fBodyGyro-std-Z  
fBodyGyro-meanFreq-X  
fBodyGyro-meanFreq-Y  
fBodyGyro-meanFreq-Z  
fBodyAccMag-mean  
fBodyAccMag-std  
fBodyAccMag-meanFreq  
fBodyBodyAccJerkMag-mean  
fBodyBodyAccJerkMag-std  
fBodyBodyAccJerkMag-meanFreq  
fBodyBodyGyroMag-mean  
fBodyBodyGyroMag-std  
fBodyBodyGyroMag-meanFreq  
fBodyBodyGyroJerkMag-mean  
fBodyBodyGyroJerkMag-std  
fBodyBodyGyroJerkMag-meanFreq