

GettingAndCleaningData_PA1

Itnava

02/18/2015

Check if “UCI HAR Dataset” directory exists and download file if it does not. Unzip file and change working directory.

```
library(knitr)
library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##   filter
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

setwd("~/Coursera/GettingAndCleaningData/Week3")
if(!file.exists("UCI\ HAR\ Dataset")){
  download.file("https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip",
  unzip("Fucidata.zip")
}
file.rename("UCI\ HAR\ Dataset", "UCI_HAR_Dataset")

## [1] TRUE

setwd("~/Coursera/GettingAndCleaningData/Week3/UCI_HAR_Dataset/")
getwd()

## [1] "/home/avanticoursera/Coursera/GettingAndCleaningData/Week3/UCI_HAR_Dataset"
```

Read the features, activity_labels and the test and train datasets as well as the labels.

```
features <- read.table("features.txt", stringsAsFactors = FALSE) activity_labels <- read.table("activity_labels.txt",
stringsAsFactors = FALSE) test <- read.table("./test/X_test.txt", stringsAsFactors = FALSE) train <-
read.table("./train/X_train.txt", stringsAsFactors = FALSE) train_labels <- read.table("./train/y_train.txt",
stringsAsFactors = FALSE) subject_test <- read.table("./test/subject_test.txt", stringsAsFactors = FALSE)
test_labels <- read.table("./test/y_test.txt") subject_train <- read.table("./train/subject_train.txt") test
<- read.table("./test/X_test.txt")
```

Combine labels with the data, separately for the test and training datasets.

```
#``{r}
test_data <- cbind(subject_test, test_labels, test)
train_data <- cbind(subject_train, train_labels, train)
```

Change column names for first two columns to “subject” and “activity” for the test and training data sets.

```
colnames(test_data) <- c("subject", "activity", as.character(features[,2])) colnames(train_data) <-  
c("subject", "activity", as.character(features[,2]))
```

Combine the training and test datasets.

```
alldata <- rbind(train_data, test_data)
```

Column names with “mean” and “std” presumably describe columns of interest, since we have to extract the mean and standard deviation of each measurement.

```
listmean <- grep("mean", colnames(alldata), ignore.case = TRUE) listdev <- grep("std", colnames(alldata),  
ignore.case = TRUE)
```

Combine the two lists and sort them. Use these to select columns of interest.

```
selectlist <- sort(append(listmean, listdev))
```

Change activity labels to names.

```
activityname <- c(alldata[[2]])  
replace_activity <- function(activityname) { activity_name = as.character(activity_labels$V2[activityname])  
}  
activity_name <- as.factor(sapply(activityname, replace_activity))
```

Insert the name in the activity column instead of code.

```
df <- tbl_df(alldata[c(1, 2, selectlist)])  
df[2] = activity_name
```

Changing activity label to something more specific. I am classifying all activities with “WALKING” as “MOVING”. “STANDING”, “SITTING” and “LAYING” are classified as “STATIONARY”.

```
classify_activity <- function(activityname) { if(activityname <=3) activity_type = “STATIONARY” else  
activity_type = “MOVING” }  
activity_type <- sapply(activityname, classify_activity)
```

Changing activity type to new classes.

```
df[2] = activity_type
```

Aggregating the mean and standard deviation for each subject and activity.

```
df_tidy <- aggregate(df[3:88], df[1:2], mean)
```

Cleaning up column names to make them compatible with R.

```
oldnames <- colnames(df_tidy)

newnames <- make.names(oldnames)
colnames(df_tidy) <- newnames
```

Writing out the tidy data table.

```
write.table(df_tidy, file = "tidy_output.txt", row.name= FALSE)
```

““