Coursera-03 - Getting and Cleaning Data, Course Project

Stanislav Gerasymenko

Sat Jan 24 15:50:52 2015

Contents

Preparing the environment **1**

Creating functions **2**

Reading in the data **2**

Subsetting, rearranging and regrouping the data **3**

Viewing and exporting the results **5**

Cleaning up the environment **8**

Preparing the environment

Clean the working environment to guarantee intended execution of the code:

```
rm(list = ls())
```

Load required libraries:

```
require(reshape2)
require(ggplot2)
require(rmarkdown)
```

Store the path to main working directory for convenience:

```
wd <- paste("C:/!SG/OneDrive/Projects/SCIENCE/Coursera/",
            "Coursera_03_Getting_and_Cleaning_Data/CourseProject/",
            sep="")
```

Creating functions

Create a function for reading a text file into a data frame:

```
readf <- function(f, names) {
        x <- read.csv(f, header = FALSE, na.strings = c(""), sep = "")
        names(x) <- names
        x
}
```

Create a function for writing a data frame into a text file:

```
writef <- function(x, f) {
        write.table(x, file = f, row.name = FALSE, quote = TRUE)
}
```

Create a function for subsetting a given data frame by matching the fragments of column names ('mean', 'std') while excluding the with 'meanFreq' and 'angle' fragments:

```
filter_names <- function(x) {
        x <- x[, (grepl("mean", names(x), ignore.case = TRUE) |
                        grepl("std", names(x), ignore.case = TRUE)) &
                        !(grepl("meanFreq", names(x), ignore.case = TRUE) |
                                grepl("angle", names(x), ignore.case = TRUE))]
        x
}
```

Create a function for reassigning the factor levels of a given variable:

```
remap_factors <- function(x, from, to) {
        for (i in 1:length(from)) {
                x[x == as.character(from[i])] <- to[i]
        }
        return(x)
}
```

Reading in the data

Read files into separate data frames and assign appropriate column names:

```
setwd(paste(wd, "Data/UCI HAR Dataset", sep="/"))

features <- readf("features.txt", "feature_name")

activity_labels <- readf("activity_labels.txt", c("activity", "label"))

setwd(paste(wd, "Data/UCI HAR Dataset/test", sep="/"))
```

```
subject_test <- readf("subject_test.txt", "subject")

X_test <- readf("X_test.txt", features[, 2])

y_test <- readf("y_test.txt", "activity")

setwd(paste(wd, "Data/UCI HAR Dataset/train", sep="/"))

subject_train <- readf("subject_train.txt", "subject")

X_train <- readf("X_train.txt", features[, 2])

y_train <- readf("y_train.txt", "activity")

# testing code:
# unique(features)
# head(X_test)
# head(y_test)
# head(subject_test)
# head(X_train)
# head(y_train)
# head(subject_train)
```

Subsetting, rearranging and regrouping the data

Select variables with 'mean' and 'std' values, combine data from all files and transform the wide table into narrow table:

```
X_test_filtered <- filter_names(X_test)

X_train_filtered <- filter_names(X_train)

# testing code:
# head(X_train_filtered)
# head(X_test_filtered)
# names(X_train_filtered)

db1 <- cbind(subject_test, y_test, X_test_filtered)

db2 <- cbind(subject_train, y_train, X_train_filtered)

# testing code:
# head(db2)
# head(db1)

db1 <- melt(db1, id = c("subject", "activity"))

db2 <- melt(db2, id = c("subject", "activity"))
```

```
db1[, 5] <- "TESTING"

db2[, 5] <- "TRAINING"

names(db1)[5] <- "mode"

names(db2)[5] <- "mode"

# testing code:
# head(db1)
# head(db2)

tidy_data <- rbind(db1, db2)

# testing code:
# head(tidy_data)
# tail(tidy_data)
# unique(tidy_data$variable)
```

Clean up the variable names by removing the '()' elements, replacing '-' with '_' and correcting the 'BodyBody' typo:

```
tidy_data[, "variable"] <- gsub("\\(\\)", "", tidy_data[, "variable"])
tidy_data[, "variable"] <- gsub("-", "_", tidy_data[, "variable"])
tidy_data[, "variable"] <- gsub("BodyBody", "Body", tidy_data[, "variable"])
```

Aassign descriptive factors to 'activity' variables, convert values of 'variable', 'mode', 'subject' and 'activity' variables to factors and reorder columns:

```
tidy_data[, "variable"] <- factor(tidy_data[, "variable"])
tidy_data[, "mode"] <- factor(tidy_data[, "mode"])
tidy_data[, "subject"] <- factor(tidy_data[, "subject"])
tidy_data[, "activity"] <- remap_factors(tidy_data[, "activity"],
                                         activity_labels[, 1],
                                         as.character(activity_labels[, 2]))
tidy_data[, "activity"] <- factor(tidy_data[, "activity"])

tidy_data <- tidy_data[, c(5, 1:4)]

# testing code:
# unique(tidy_data$variable)
# str(tidy_data)
# head(tidy_data)
# tail(tidy_data)
# ggplot(tidy_data, aes(value, variable, color=value)) +
#         geom_point() +
#         facet_grid(. ~ activity)
```

Summarize the data by three columns (subject, activity, variable) applying the mean function to 'value' column:

```
tidy_data_averaged <- aggregate(tidy_data$value,
                                by = list(tidy_data$subject,
                                          tidy_data$activity,
                                          tidy_data$variable),
                                FUN = mean)
```

Reassign the column names and add the 'meanOf_' description to variable names:

```
names(tidy_data_averaged) <- c("subject", "activity", "variable", "value")

tidy_data_averaged[, "variable"] <- gsub("^",
                                         "meanOf_",
                                         tidy_data_averaged[, "variable"])
tidy_data_averaged[, "variable"] <- factor(tidy_data_averaged[, "variable"])
```

Viewing and exporting the results

```
unique(tidy_data_averaged$variable)
```

```
##  [1] meanOffBodyAcc-mean-X       meanOffBodyAcc-mean-Y
##  [3] meanOffBodyAcc-mean-Z       meanOffBodyAcc-std-X
##  [5] meanOffBodyAcc-std-Y        meanOffBodyAcc-std-Z
##  [7] meanOffBodyAccJerk-mean-X   meanOffBodyAccJerk-mean-Y
##  [9] meanOffBodyAccJerk-mean-Z   meanOffBodyAccJerk-std-X
## [11] meanOffBodyAccJerk-std-Y    meanOffBodyAccJerk-std-Z
## [13] meanOffBodyAccJerkMag-mean  meanOffBodyAccJerkMag-std
## [15] meanOffBodyAccMag-mean      meanOffBodyAccMag-std
## [17] meanOffBodyGyro-mean-X      meanOffBodyGyro-mean-Y
## [19] meanOffBodyGyro-mean-Z      meanOffBodyGyro-std-X
## [21] meanOffBodyGyro-std-Y       meanOffBodyGyro-std-Z
## [23] meanOffBodyGyroJerkMag-mean meanOffBodyGyroJerkMag-std
## [25] meanOffBodyGyroMag-mean     meanOffBodyGyroMag-std
## [27] meanOftBodyAcc-mean-X       meanOftBodyAcc-mean-Y
## [29] meanOftBodyAcc-mean-Z       meanOftBodyAcc-std-X
## [31] meanOftBodyAcc-std-Y        meanOftBodyAcc-std-Z
## [33] meanOftBodyAccJerk-mean-X   meanOftBodyAccJerk-mean-Y
## [35] meanOftBodyAccJerk-mean-Z   meanOftBodyAccJerk-std-X
## [37] meanOftBodyAccJerk-std-Y    meanOftBodyAccJerk-std-Z
## [39] meanOftBodyAccJerkMag-mean  meanOftBodyAccJerkMag-std
## [41] meanOftBodyAccMag-mean      meanOftBodyAccMag-std
## [43] meanOftBodyGyro-mean-X      meanOftBodyGyro-mean-Y
## [45] meanOftBodyGyro-mean-Z      meanOftBodyGyro-std-X
## [47] meanOftBodyGyro-std-Y       meanOftBodyGyro-std-Z
## [49] meanOftBodyGyroJerk-mean-X  meanOftBodyGyroJerk-mean-Y
## [51] meanOftBodyGyroJerk-mean-Z  meanOftBodyGyroJerk-std-X
## [53] meanOftBodyGyroJerk-std-Y   meanOftBodyGyroJerk-std-Z
## [55] meanOftBodyGyroJerkMag-mean meanOftBodyGyroJerkMag-std
## [57] meanOftBodyGyroMag-mean     meanOftBodyGyroMag-std
```

```
## [59] meanOftGravityAcc-mean-X    meanOftGravityAcc-mean-Y
## [61] meanOftGravityAcc-mean-Z    meanOftGravityAcc-std-X
## [63] meanOftGravityAcc-std-Y     meanOftGravityAcc-std-Z
## [65] meanOftGravityAccMag-mean   meanOftGravityAccMag-std
## 66 Levels: meanOffBodyAcc-mean-X ... meanOftGravityAccMag-std
```

```
str(tidy_data_averaged)
```

```
## 'data.frame':    11880 obs. of  4 variables:
##  $ subject : Factor w/ 30 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ activity: Factor w/ 6 levels "LAYING","SITTING",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ variable: Factor w/ 66 levels "meanOffBodyAcc-mean-X",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ value   : num  -0.939 -0.977 -0.981 -0.959 -0.969 ...
```

```
head(tidy_data_averaged)
```

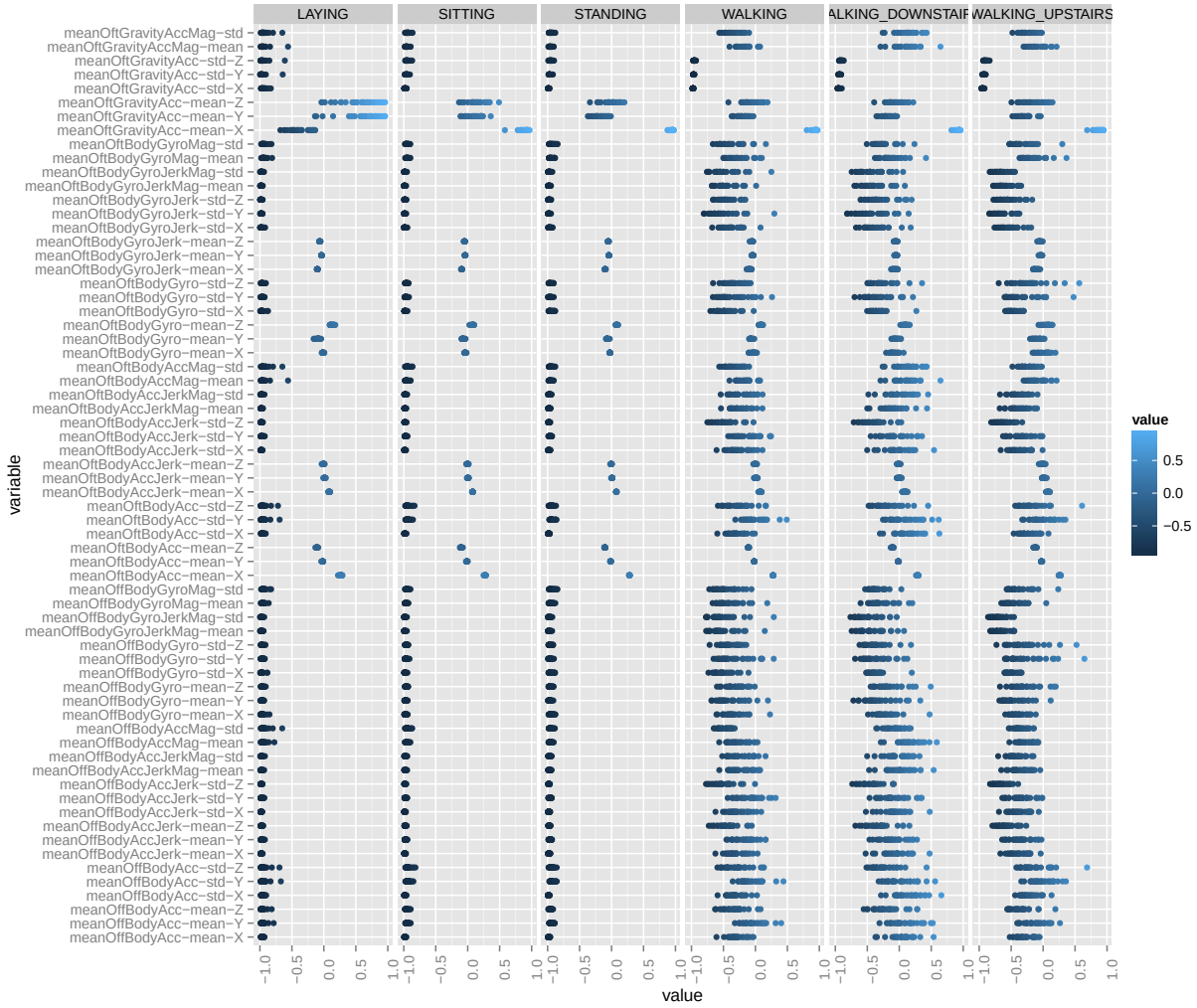
```
##   subject activity              variable      value
## 1       1   LAYING meanOffBodyAcc-mean-X -0.9390991
## 2       2   LAYING meanOffBodyAcc-mean-X -0.9767251
## 3       3   LAYING meanOffBodyAcc-mean-X -0.9806656
## 4       4   LAYING meanOffBodyAcc-mean-X -0.9588021
## 5       5   LAYING meanOffBodyAcc-mean-X -0.9687417
## 6       6   LAYING meanOffBodyAcc-mean-X -0.9391143
```

```
tail(tidy_data_averaged)
```

```
##       subject         activity                 variable       value
## 11875      25 WALKING_UPSTAIRS meanOftGravityAccMag-std -0.48000665
## 11876      26 WALKING_UPSTAIRS meanOftGravityAccMag-std -0.15070711
## 11877      27 WALKING_UPSTAIRS meanOftGravityAccMag-std -0.37994879
## 11878      28 WALKING_UPSTAIRS meanOftGravityAccMag-std -0.21192867
## 11879      29 WALKING_UPSTAIRS meanOftGravityAccMag-std -0.04146961
## 11880      30 WALKING_UPSTAIRS meanOftGravityAccMag-std -0.32741082
```

Plots:


meanOftGravityAccMag−std
meanOftGravityAccMag−mean
meanOftGravityAcc−std−Z
meanOftGravityAcc−std−Y
meanOftGravityAcc−std−X
meanOftGravityAcc−mean−Z
meanOftGravityAcc−mean−Y
meanOftGravityAcc−mean−X
meanOftBodyGyroMag−std
meanOftBodyGyroMag−mean
meanOftBodyGyroJerkMag−std
meanOftBodyGyroJerkMag−mean
meanOftBodyGyroJerk−std−Z
meanOftBodyGyroJerk−std−Y
meanOftBodyGyroJerk−std−X
meanOftBodyGyroJerk−mean−Z
meanOftBodyGyroJerk−mean−Y
meanOftBodyGyroJerk−mean−X
meanOftBodyGyro−std−Z
meanOftBodyGyro−std−Y
meanOftBodyGyro−std−X
meanOftBodyGyro−mean−Z
meanOftBodyGyro−mean−Y
meanOftBodyGyro−mean−X
meanOftBodyAccMag−std
meanOftBodyAccMag−mean
meanOftBodyAccJerkMag−std
meanOftBodyAccJerkMag−mean
meanOftBodyAccJerk−std−Z
meanOftBodyAccJerk−std−Y
meanOftBodyAccJerk−std−X
meanOftBodyAccJerk−mean−Z
meanOftBodyAccJerk−mean−Y
meanOftBodyAccJerk−mean−X
meanOftBodyAcc−std−Z
meanOftBodyAcc−std−Y
meanOftBodyAcc−std−X
meanOftBodyAcc−mean−Z
meanOftBodyAcc−mean−Y
meanOftBodyAcc−mean−X
meanOffBodyGyroMag−std
meanOffBodyGyroMag−mean
meanOffBodyGyroJerkMag−std
meanOffBodyGyroJerkMag−mean
meanOffBodyGyro−std−Z
meanOffBodyGyro−std−Y
meanOffBodyGyro−std−X
meanOffBodyGyro−mean−Z
meanOffBodyGyro−mean−Y
meanOffBodyGyro−mean−X
meanOffBodyAccMag−std
meanOffBodyAccMag−mean
meanOffBodyAccJerkMag−std
meanOffBodyAccJerkMag−mean
meanOffBodyAccJerk−std−Z
meanOffBodyAccJerk−std−Y
meanOffBodyAccJerk−std−X
meanOffBodyAccJerk−mean−Z
meanOffBodyAccJerk−mean−Y
meanOffBodyAccJerk−mean−X
meanOffBodyAcc−std−Z
meanOffBodyAcc−std−Y
meanOffBodyAcc−std−X
meanOffBodyAcc−mean−Z
meanOffBodyAcc−mean−Y
meanOffBodyAcc−mean−X
variable
−1.0
−0.5
0.0
0.5
1.0
value
value
0.5
0.0
−0.5



Write the tidy dataset and the summarized tidy dataset into separate text files:

```
setwd(paste(wd, "Data/Tidy_data", sep="/"))

writef(tidy_data, "tidy_data.txt")

writef(tidy_data_averaged, "tidy_data_averaged.txt")

# The files may be read back into R using the following code:
# read.csv("filenema.txt", header = TRUE, sep = "")
```

Cleaning up the environment

Set the working directory, in which the R session files should be recorded:

```
setwd(wd)
```

Remove all the temporary objects, leaving only two final tidy datasets:

```
rm(list = setdiff(ls(), c("tidy_data", "tidy_data_averaged")))
```