Cleaning_And_Tidying

Manish Gyawali

December 18, 2018

ABOUT THE PROGRAM

This program takes as its input the various datasets that we have been given, performs the necessary operations on them, and finally produces as output a cleaned, tidied dataset.

DATASETS GIVEN:

features.txt
561-feature vector describing different motion statistics. Only the 79 feature vector describing statistics related to mean/std are relevant to us

activity_labels.txt labels the 6 kinds of motion that subjects did

X_test.txt X_train.txt Observations of subjects belonging to test and train group, respectively in terms of the of 561-variable vector of statistics .

y_test.txt y_train.txt Observations of test,train subjects in terms of one of their 6 movements.These are 1-variable vectors.

subject_test.txt subject_train.txt Observations of test,train subjects in terms of one of the 30 index numbers they have been alloted. These are 1-variable vectors.

DATASETS CREATED BY ME:

merged_training_set – Combination of X-test.txt,X_train.txt

merged_movement – Combination of y-test.txt,y_train.txt

all_Persons – Combination of subject-test.txt,subject_train.txt

labelled_Movements – merges the 561 vector dataset for all 10299 observations with their activity labels so that each observation has an associated activity label

locations – defines the locations on the 561 dimension feature vector in which either mean, standard deviation and nothing else are present. locations is thus a 79 dimension vector

req_features – This is a 79 dimension vector that gives the names of the 79 statistics corresponding to locations.

b1 – This stores the names of req_features in the proper format

a1 – Our merged_training_set was a 10299 X 561 dataframe that gave us all the statistics for all observations. a1 removes the unnecessary (i.e non mean/std) statistics to give us only a 10299 X 79 dataframe

a2 – This merges the index numbers of the persons with a1. Now we have a dataframe that has all observations, and in which all PERSONS are accounted for. We now have a 10299 X 80 dimension dataframe.

c2 – This merges activity description to a2. Now we have all observations, all persons accounted for, and all ACTIVITIES accounted for.This is our final, 10299 X 81 dimension dataframe

summ1 – summarizes the means for c2, i.e: all people, all activities ordered_summ1 – orders summ1 according to peoples' index number

```r
setwd("E:/Coursera/Getting_Cleaning_Data/Final/UCI")

features <- read.csv("features.txt", sep = "", header = FALSE, stringsAsFactors = F)
```

```r
activity_labels <- read.csv("activity_labels.txt", sep = "", header = FALSE )
testSet <- read.csv("X_test.txt", sep = "", header = FALSE)
trainSet <- read.csv("X_train.txt", sep = "", header = FALSE)
merged_training_test <- rbind(testSet,trainSet) #combination of testSet, trainSet
testMoves <- read.csv("y_test.txt", sep = "", header = FALSE) #2947 by 1 matrix. Columns
trainMoves <- read.csv("y_train.txt", sep = "", header = FALSE)
merged_Movement <- rbind(testMoves, trainMoves)
testPerson <- read.csv("subject_test.txt", sep = "", header = FALSE)
trainPerson <- read.csv("subject_train.txt", sep = "", header = FALSE)
all_Persons <- rbind(testPerson, trainPerson)
labelled_Movements <- merge(merged_Movement, activity_labels)
locations = grep(pattern = "-mean|-std", x = features[,c(2)])   #1 by 79 Matrix of locations of require
req_features <- grep(pattern = "-mean|-std", x = features[,c(2)], value = T) #1 by 79 Matrix of require
a1 =  merged_training_test[locations]                #Creates a dataframe a1 that applies values of loc
#dataframe merged_training_test (10299 by 561) creating a (10299 by 79) dataframe
b1 <- t(req_features)                           # Creates a chr dataframe b1 that stores the names of t
a2 <- bind_cols(all_Persons, a1)                    # Binds a1(required features) with all_Persons(set
c2 <- bind_cols(labelled_Movements[2], a2)       # binds a2 with activity description to create complet
names(c2)[1] = "Activity"
names(c2)[2] = "Person_Number"

#Extractng the Means of all the variables for all Persons and all Activities
summ1 <- summarise_each(group_by(c2, Activity, Person_Number ), funs(mean))
```

```
## `summarise_each()` is deprecated.
## Use `summarise_all()`, `summarise_at()` or `summarise_if()` instead.
## To map `funs` over all variables, use `summarise_all()`
```

```r
ordered_summ1 <- summ1[order(summ1$Person_Number),]

#OUTPUT

write.table(summ1[order(summ1$Person_Number),] , file = "Clean_Tidy.txt", row.names = FALSE)
```