**This Run analysis file describes the codes to create dataset2 output**

**Instructions for project**

The purpose of this project is to demonstrate your ability to collect, work with, and clean a data set. The goal is to prepare tidy data that can be used for later analysis. You will be graded by your peers on a series of yes/no questions related to the project. You will be required to submit: 1) a tidy data set as described below, 2) a link to a Github repository with your script for performing the analysis, and 3) a code book that describes the variables, the data, and any transformations or work that you performed to clean up the data called CodeBook.md. You should also include a README.md in the repo with your scripts. This repo explains how all of the scripts work and how they are connected.

One of the most exciting areas in all of data science right now is wearable computing - see for example this article . Companies like Fitbit, Nike, and Jawbone Up are racing to develop the most advanced algorithms to attract new users. The data linked to from the course website represent data collected from the accelerometers from the Samsung Galaxy S smartphone. A full description is available at the site where the data was obtained:

<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

Here are the data for the project:

<https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip>

You should create one R script called run\_analysis.R that does the following.

1. Merges the training and the test sets to create one data set.
2. Extracts only the measurements on the mean and standard deviation for each measurement.
3. Uses descriptive activity names to name the activities in the data set
4. Appropriately labels the data set with descriptive variable names.
5. From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.

**Get the data**

1.**Download the file and put the file in the folder**

**if**(!file.exists("./data")){dir.create("./data")}

fileUrl <- "https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"

download.file(fileUrl,destfile="./data/Dataset.zip",method="curl")

2.**Unzip the file**

unzip(zipfile="./data/Dataset.zip",exdir="./data")

3.**unzipped files are in the folderUCI HAR Dataset. Get the list of the files**

path\_rf <- file.path("./data" , "UCI HAR Dataset")

files<-list.files(path\_rf, recursive=TRUE)

files

## [1] "activity\_labels.txt"

## [2] "features\_info.txt"

## [3] "features.txt"

## [4] "README.txt"

## [5] "test/Inertial Signals/body\_acc\_x\_test.txt"

## [6] "test/Inertial Signals/body\_acc\_y\_test.txt"

## [7] "test/Inertial Signals/body\_acc\_z\_test.txt"

## [8] "test/Inertial Signals/body\_gyro\_x\_test.txt"

## [9] "test/Inertial Signals/body\_gyro\_y\_test.txt"

## [10] "test/Inertial Signals/body\_gyro\_z\_test.txt"

## [11] "test/Inertial Signals/total\_acc\_x\_test.txt"

## [12] "test/Inertial Signals/total\_acc\_y\_test.txt"

## [13] "test/Inertial Signals/total\_acc\_z\_test.txt"

## [14] "test/subject\_test.txt"

## [15] "test/X\_test.txt"

## [16] "test/y\_test.txt"

## [17] "train/Inertial Signals/body\_acc\_x\_train.txt"

## [18] "train/Inertial Signals/body\_acc\_y\_train.txt"

## [19] "train/Inertial Signals/body\_acc\_z\_train.txt"

## [20] "train/Inertial Signals/body\_gyro\_x\_train.txt"

## [21] "train/Inertial Signals/body\_gyro\_y\_train.txt"

## [22] "train/Inertial Signals/body\_gyro\_z\_train.txt"

## [23] "train/Inertial Signals/total\_acc\_x\_train.txt"

## [24] "train/Inertial Signals/total\_acc\_y\_train.txt"

## [25] "train/Inertial Signals/total\_acc\_z\_train.txt"

## [26] "train/subject\_train.txt"

## [27] "train/X\_train.txt"

## [28] "train/y\_train.txt"

**See the README.txt file for the detailed information on the dataset.** **For the purposes of this project, the files in the Inertial Signals folders are not used.** **The files that will be used to load data are listed as follows:**

* test/subject\_test.txt
* test/X\_test.txt
* test/y\_test.txt
* train/subject\_train.txt
* train/X\_train.txt
* train/y\_train.txt

**Read data from the targeted files**

1. **Get the big picture of the structure of the data frame that will be used in this project.**

Reference link: <https://class.coursera.org/getdata-008/forum/thread?thread_id=24>

From the related files, we can see:

1. Values of Varible Activity consist of data from “Y\_train.txt” and “Y\_test.txt”
2. values of Varible Subject consist of data from “subject\_train.txt” and subject\_test.txt"
3. Values of Varibles Features consist of data from “X\_train.txt” and “X\_test.txt”
4. Names of Varibles Features come from “features.txt”
5. levels of Varible Activity come from “activity\_labels.txt”

So we will use Activity, Subject and Features as part of descriptive variable names for data in data frame.

2.**Read data from the files into the variables**

Read the Activity files

dataActivityTest <- read.table(file.path(path\_rf, "test" , "Y\_test.txt" ),header = FALSE)

dataActivityTrain <- read.table(file.path(path\_rf, "train", "Y\_train.txt"),header = FALSE)

Read the Subject files

dataSubjectTrain <- read.table(file.path(path\_rf, "train", "subject\_train.txt"),header = FALSE)

dataSubjectTest <- read.table(file.path(path\_rf, "test" , "subject\_test.txt"),header = FALSE)

Read Fearures files

dataFeaturesTest <- read.table(file.path(path\_rf, "test" , "X\_test.txt" ),header = FALSE)

dataFeaturesTrain <- read.table(file.path(path\_rf, "train", "X\_train.txt"),header = FALSE)

1. **Look at the properties of the above varibles**

str(dataActivityTest)

## 'data.frame': 2947 obs. of 1 variable:

## $ V1: int 5 5 5 5 5 5 5 5 5 5 ...

str(dataActivityTrain)

## 'data.frame': 7352 obs. of 1 variable:

## $ V1: int 5 5 5 5 5 5 5 5 5 5 ...

str(dataSubjectTrain)

## 'data.frame': 7352 obs. of 1 variable:

## $ V1: int 1 1 1 1 1 1 1 1 1 1 ...

str(dataSubjectTest)

## 'data.frame': 2947 obs. of 1 variable:

## $ V1: int 2 2 2 2 2 2 2 2 2 2 ...

str(dataFeaturesTest)

## 'data.frame': 2947 obs. of 561 variables:

## $ V1 : num 0.257 0.286 0.275 0.27 0.275 ...

## $ V2 : num -0.0233 -0.0132 -0.0261 -0.0326 -0.0278 ...

## $ V3 : num -0.0147 -0.1191 -0.1182 -0.1175 -0.1295 ...

## $ V4 : num -0.938 -0.975 -0.994 -0.995 -0.994 ...

## $ V5 : num -0.92 -0.967 -0.97 -0.973 -0.967 ...

## $ V6 : num -0.668 -0.945 -0.963 -0.967 -0.978 ...

## $ V7 : num -0.953 -0.987 -0.994 -0.995 -0.994 ...

## $ V8 : num -0.925 -0.968 -0.971 -0.974 -0.966 ...

## $ V9 : num -0.674 -0.946 -0.963 -0.969 -0.977 ...

## $ V10 : num -0.894 -0.894 -0.939 -0.939 -0.939 ...

## $ V11 : num -0.555 -0.555 -0.569 -0.569 -0.561 ...

## $ V12 : num -0.466 -0.806 -0.799 -0.799 -0.826 ...

## $ V13 : num 0.717 0.768 0.848 0.848 0.849 ...

## $ V14 : num 0.636 0.684 0.668 0.668 0.671 ...

## $ V15 : num 0.789 0.797 0.822 0.822 0.83 ...

## $ V16 : num -0.878 -0.969 -0.977 -0.974 -0.975 ...

## $ V17 : num -0.998 -1 -1 -1 -1 ...

## $ V18 : num -0.998 -1 -1 -0.999 -0.999 ...

## $ V19 : num -0.934 -0.998 -0.999 -0.999 -0.999 ...

## $ V20 : num -0.976 -0.994 -0.993 -0.995 -0.993 ...

## $ V21 : num -0.95 -0.974 -0.974 -0.979 -0.967 ...

## $ V22 : num -0.83 -0.951 -0.965 -0.97 -0.976 ...

## $ V23 : num -0.168 -0.302 -0.618 -0.75 -0.591 ...

## $ V24 : num -0.379 -0.348 -0.695 -0.899 -0.74 ...

## $ V25 : num 0.246 -0.405 -0.537 -0.554 -0.799 ...

## $ V26 : num 0.521 0.507 0.242 0.175 0.116 ...

## $ V27 : num -0.4878 -0.1565 -0.115 -0.0513 -0.0289 ...

## $ V28 : num 0.4823 0.0407 0.0327 0.0342 -0.0328 ...

## $ V29 : num -0.0455 0.273 0.1924 0.1536 0.2943 ...

## $ V30 : num 0.21196 0.19757 -0.01194 0.03077 0.00063 ...

## $ V31 : num -0.1349 -0.1946 -0.0634 -0.1293 -0.0453 ...

## $ V32 : num 0.131 0.411 0.471 0.446 0.168 ...

## $ V33 : num -0.0142 -0.3405 -0.5074 -0.4195 -0.0682 ...

## $ V34 : num -0.106 0.0776 0.1885 0.2715 0.0744 ...

## $ V35 : num 0.0735 -0.084 -0.2316 -0.2258 0.0271 ...

## $ V36 : num -0.1715 0.0353 0.6321 0.4164 -0.1459 ...

## $ V37 : num 0.0401 -0.0101 -0.5507 -0.2864 -0.0502 ...

## $ V38 : num 0.077 -0.105 0.3057 -0.0638 0.2352 ...

## $ V39 : num -0.491 -0.429 -0.324 -0.167 0.29 ...

## $ V40 : num -0.709 0.399 0.28 0.545 0.458 ...

## $ V41 : num 0.936 0.927 0.93 0.929 0.927 ...

## $ V42 : num -0.283 -0.289 -0.288 -0.293 -0.303 ...

## $ V43 : num 0.115 0.153 0.146 0.143 0.138 ...

## $ V44 : num -0.925 -0.989 -0.996 -0.993 -0.996 ...

## $ V45 : num -0.937 -0.984 -0.988 -0.97 -0.971 ...

## $ V46 : num -0.564 -0.965 -0.982 -0.992 -0.968 ...

## $ V47 : num -0.93 -0.989 -0.996 -0.993 -0.996 ...

## $ V48 : num -0.938 -0.983 -0.989 -0.971 -0.971 ...

## $ V49 : num -0.606 -0.965 -0.98 -0.993 -0.969 ...

## $ V50 : num 0.906 0.856 0.856 0.856 0.854 ...

## $ V51 : num -0.279 -0.305 -0.305 -0.305 -0.313 ...

## $ V52 : num 0.153 0.153 0.139 0.136 0.134 ...

## $ V53 : num 0.944 0.944 0.949 0.947 0.946 ...

## $ V54 : num -0.262 -0.262 -0.262 -0.273 -0.279 ...

## $ V55 : num -0.0762 0.149 0.145 0.1421 0.1309 ...

## $ V56 : num -0.0178 0.0577 0.0406 0.0461 0.0554 ...

## $ V57 : num 0.829 0.806 0.812 0.809 0.804 ...

## $ V58 : num -0.865 -0.858 -0.86 -0.854 -0.843 ...

## $ V59 : num -0.968 -0.957 -0.961 -0.963 -0.965 ...

## $ V60 : num -0.95 -0.988 -0.996 -0.992 -0.996 ...

## $ V61 : num -0.946 -0.982 -0.99 -0.973 -0.972 ...

## $ V62 : num -0.76 -0.971 -0.979 -0.996 -0.969 ...

## $ V63 : num -0.425 -0.729 -0.823 -0.823 -0.83 ...

## $ V64 : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

## $ V65 : num 0.219 -0.465 -0.53 -0.7 -0.302 ...

## $ V66 : num -0.43 -0.51 -0.295 -0.343 -0.482 ...

## $ V67 : num 0.431 0.525 0.305 0.359 0.539 ...

## $ V68 : num -0.432 -0.54 -0.315 -0.375 -0.596 ...

## $ V69 : num 0.433 0.554 0.326 0.392 0.655 ...

## $ V70 : num -0.795 -0.746 -0.232 -0.233 -0.493 ...

## $ V71 : num 0.781 0.733 0.169 0.176 0.463 ...

## $ V72 : num -0.78 -0.737 -0.155 -0.169 -0.465 ...

## $ V73 : num 0.785 0.749 0.164 0.185 0.483 ...

## $ V74 : num -0.984 -0.845 -0.429 -0.297 -0.536 ...

## $ V75 : num 0.987 0.869 0.44 0.304 0.544 ...

## $ V76 : num -0.989 -0.893 -0.451 -0.311 -0.553 ...

## $ V77 : num 0.988 0.913 0.458 0.315 0.559 ...

## $ V78 : num 0.981 0.945 0.548 0.986 0.998 ...

## $ V79 : num -0.996 -0.911 -0.335 0.653 0.916 ...

## $ V80 : num -0.96 -0.739 0.59 0.747 0.929 ...

## $ V81 : num 0.072 0.0702 0.0694 0.0749 0.0784 ...

## $ V82 : num 0.04575 -0.01788 -0.00491 0.03227 0.02228 ...

## $ V83 : num -0.10604 -0.00172 -0.01367 0.01214 0.00275 ...

## $ V84 : num -0.907 -0.949 -0.991 -0.991 -0.992 ...

## $ V85 : num -0.938 -0.973 -0.971 -0.973 -0.979 ...

## $ V86 : num -0.936 -0.978 -0.973 -0.976 -0.987 ...

## $ V87 : num -0.916 -0.969 -0.991 -0.99 -0.991 ...

## $ V88 : num -0.937 -0.974 -0.973 -0.973 -0.977 ...

## $ V89 : num -0.949 -0.979 -0.975 -0.978 -0.985 ...

## $ V90 : num -0.903 -0.915 -0.992 -0.992 -0.994 ...

## $ V91 : num -0.95 -0.981 -0.975 -0.975 -0.986 ...

## $ V92 : num -0.891 -0.978 -0.962 -0.962 -0.986 ...

## $ V93 : num 0.898 0.898 0.994 0.994 0.994 ...

## $ V94 : num 0.95 0.968 0.976 0.976 0.98 ...

## $ V95 : num 0.946 0.966 0.966 0.97 0.985 ...

## $ V96 : num -0.931 -0.974 -0.982 -0.983 -0.987 ...

## $ V97 : num -0.995 -0.998 -1 -1 -1 ...

## $ V98 : num -0.997 -0.999 -0.999 -0.999 -1 ...

## $ V99 : num -0.997 -0.999 -0.999 -0.999 -1 ...

## [list output truncated]

str(dataFeaturesTrain)

## 'data.frame': 7352 obs. of 561 variables:

## $ V1 : num 0.289 0.278 0.28 0.279 0.277 ...

## $ V2 : num -0.0203 -0.0164 -0.0195 -0.0262 -0.0166 ...

## $ V3 : num -0.133 -0.124 -0.113 -0.123 -0.115 ...

## $ V4 : num -0.995 -0.998 -0.995 -0.996 -0.998 ...

## $ V5 : num -0.983 -0.975 -0.967 -0.983 -0.981 ...

## $ V6 : num -0.914 -0.96 -0.979 -0.991 -0.99 ...

## $ V7 : num -0.995 -0.999 -0.997 -0.997 -0.998 ...

## $ V8 : num -0.983 -0.975 -0.964 -0.983 -0.98 ...

## $ V9 : num -0.924 -0.958 -0.977 -0.989 -0.99 ...

## $ V10 : num -0.935 -0.943 -0.939 -0.939 -0.942 ...

## $ V11 : num -0.567 -0.558 -0.558 -0.576 -0.569 ...

## $ V12 : num -0.744 -0.818 -0.818 -0.83 -0.825 ...

## $ V13 : num 0.853 0.849 0.844 0.844 0.849 ...

## $ V14 : num 0.686 0.686 0.682 0.682 0.683 ...

## $ V15 : num 0.814 0.823 0.839 0.838 0.838 ...

## $ V16 : num -0.966 -0.982 -0.983 -0.986 -0.993 ...

## $ V17 : num -1 -1 -1 -1 -1 ...

## $ V18 : num -1 -1 -1 -1 -1 ...

## $ V19 : num -0.995 -0.998 -0.999 -1 -1 ...

## $ V20 : num -0.994 -0.999 -0.997 -0.997 -0.998 ...

## $ V21 : num -0.988 -0.978 -0.965 -0.984 -0.981 ...

## $ V22 : num -0.943 -0.948 -0.975 -0.986 -0.991 ...

## $ V23 : num -0.408 -0.715 -0.592 -0.627 -0.787 ...

## $ V24 : num -0.679 -0.501 -0.486 -0.851 -0.559 ...

## $ V25 : num -0.602 -0.571 -0.571 -0.912 -0.761 ...

## $ V26 : num 0.9293 0.6116 0.273 0.0614 0.3133 ...

## $ V27 : num -0.853 -0.3295 -0.0863 0.0748 -0.1312 ...

## $ V28 : num 0.36 0.284 0.337 0.198 0.191 ...

## $ V29 : num -0.0585 0.2846 -0.1647 -0.2643 0.0869 ...

## $ V30 : num 0.2569 0.1157 0.0172 0.0725 0.2576 ...

## $ V31 : num -0.2248 -0.091 -0.0745 -0.1553 -0.2725 ...

## $ V32 : num 0.264 0.294 0.342 0.323 0.435 ...

## $ V33 : num -0.0952 -0.2812 -0.3326 -0.1708 -0.3154 ...

## $ V34 : num 0.279 0.086 0.239 0.295 0.44 ...

## $ V35 : num -0.4651 -0.0222 -0.1362 -0.3061 -0.2691 ...

## $ V36 : num 0.4919 -0.0167 0.1739 0.4821 0.1794 ...

## $ V37 : num -0.191 -0.221 -0.299 -0.47 -0.089 ...

## $ V38 : num 0.3763 -0.0134 -0.1247 -0.3057 -0.1558 ...

## $ V39 : num 0.4351 -0.0727 -0.1811 -0.3627 -0.1898 ...

## $ V40 : num 0.661 0.579 0.609 0.507 0.599 ...

## $ V41 : num 0.963 0.967 0.967 0.968 0.968 ...

## $ V42 : num -0.141 -0.142 -0.142 -0.144 -0.149 ...

## $ V43 : num 0.1154 0.1094 0.1019 0.0999 0.0945 ...

## $ V44 : num -0.985 -0.997 -1 -0.997 -0.998 ...

## $ V45 : num -0.982 -0.989 -0.993 -0.981 -0.988 ...

## $ V46 : num -0.878 -0.932 -0.993 -0.978 -0.979 ...

## $ V47 : num -0.985 -0.998 -1 -0.996 -0.998 ...

## $ V48 : num -0.984 -0.99 -0.993 -0.981 -0.989 ...

## $ V49 : num -0.895 -0.933 -0.993 -0.978 -0.979 ...

## $ V50 : num 0.892 0.892 0.892 0.894 0.894 ...

## $ V51 : num -0.161 -0.161 -0.164 -0.164 -0.167 ...

## $ V52 : num 0.1247 0.1226 0.0946 0.0934 0.0917 ...

## $ V53 : num 0.977 0.985 0.987 0.987 0.987 ...

## $ V54 : num -0.123 -0.115 -0.115 -0.121 -0.122 ...

## $ V55 : num 0.0565 0.1028 0.1028 0.0958 0.0941 ...

## $ V56 : num -0.375 -0.383 -0.402 -0.4 -0.4 ...

## $ V57 : num 0.899 0.908 0.909 0.911 0.912 ...

## $ V58 : num -0.971 -0.971 -0.97 -0.969 -0.967 ...

## $ V59 : num -0.976 -0.979 -0.982 -0.982 -0.984 ...

## $ V60 : num -0.984 -0.999 -1 -0.996 -0.998 ...

## $ V61 : num -0.989 -0.99 -0.992 -0.981 -0.991 ...

## $ V62 : num -0.918 -0.942 -0.993 -0.98 -0.98 ...

## $ V63 : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

## $ V64 : num -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

## $ V65 : num 0.114 -0.21 -0.927 -0.596 -0.617 ...

## $ V66 : num -0.59042 -0.41006 0.00223 -0.06493 -0.25727 ...

## $ V67 : num 0.5911 0.4139 0.0275 0.0754 0.2689 ...

## $ V68 : num -0.5918 -0.4176 -0.0567 -0.0858 -0.2807 ...

## $ V69 : num 0.5925 0.4213 0.0855 0.0962 0.2926 ...

## $ V70 : num -0.745 -0.196 -0.329 -0.295 -0.167 ...

## $ V71 : num 0.7209 0.1253 0.2705 0.2283 0.0899 ...

## $ V72 : num -0.7124 -0.1056 -0.2545 -0.2063 -0.0663 ...

## $ V73 : num 0.7113 0.1091 0.2576 0.2048 0.0671 ...

## $ V74 : num -0.995 -0.834 -0.705 -0.385 -0.237 ...

## $ V75 : num 0.996 0.834 0.714 0.386 0.239 ...

## $ V76 : num -0.996 -0.834 -0.723 -0.387 -0.241 ...

## $ V77 : num 0.992 0.83 0.729 0.385 0.241 ...

## $ V78 : num 0.57 -0.831 -0.181 -0.991 -0.408 ...

## $ V79 : num 0.439 -0.866 0.338 -0.969 -0.185 ...

## $ V80 : num 0.987 0.974 0.643 0.984 0.965 ...

## $ V81 : num 0.078 0.074 0.0736 0.0773 0.0734 ...

## $ V82 : num 0.005 0.00577 0.0031 0.02006 0.01912 ...

## $ V83 : num -0.06783 0.02938 -0.00905 -0.00986 0.01678 ...

## $ V84 : num -0.994 -0.996 -0.991 -0.993 -0.996 ...

## $ V85 : num -0.988 -0.981 -0.981 -0.988 -0.988 ...

## $ V86 : num -0.994 -0.992 -0.99 -0.993 -0.992 ...

## $ V87 : num -0.994 -0.996 -0.991 -0.994 -0.997 ...

## $ V88 : num -0.986 -0.979 -0.979 -0.986 -0.987 ...

## $ V89 : num -0.993 -0.991 -0.987 -0.991 -0.991 ...

## $ V90 : num -0.985 -0.995 -0.987 -0.987 -0.997 ...

## $ V91 : num -0.992 -0.979 -0.979 -0.992 -0.992 ...

## $ V92 : num -0.993 -0.992 -0.992 -0.99 -0.99 ...

## $ V93 : num 0.99 0.993 0.988 0.988 0.994 ...

## $ V94 : num 0.992 0.992 0.992 0.993 0.993 ...

## $ V95 : num 0.991 0.989 0.989 0.993 0.986 ...

## $ V96 : num -0.994 -0.991 -0.988 -0.993 -0.994 ...

## $ V97 : num -1 -1 -1 -1 -1 ...

## $ V98 : num -1 -1 -1 -1 -1 ...

## $ V99 : num -1 -1 -1 -1 -1 ...

## [list output truncated]

**Merges the training and the test sets to create one data set**

1.**Concatenate the data tables by rows**

dataSubject <- rbind(dataSubjectTrain, dataSubjectTest)

dataActivity<- rbind(dataActivityTrain, dataActivityTest)

dataFeatures<- rbind(dataFeaturesTrain, dataFeaturesTest)

2.**set names to variables**

names(dataSubject)<-c("subject")

names(dataActivity)<- c("activity")

dataFeaturesNames <- read.table(file.path(path\_rf, "features.txt"),head=FALSE)

names(dataFeatures)<- dataFeaturesNames$V2

3.**Merge columns to get the data frame Data for all data**

dataCombine <- cbind(dataSubject, dataActivity)

Data <- cbind(dataFeatures, dataCombine)

**Extracts only the measurements on the mean and standard deviation for each measurement**

1. **Subset Name of Features by measurements on the mean and standard deviation**

i.e taken Names of Features with “mean()” or “std()”

subdataFeaturesNames<-dataFeaturesNames$V2[grep("mean\\(\\)|std\\(\\)", dataFeaturesNames$V2)]

1. **Subset the data frame Data by seleted names of Features**

selectedNames<-c(as.character(subdataFeaturesNames), "subject", "activity" )

Data<-subset(Data,select=selectedNames)

**Uses descriptive activity names to name the activities in the data set**

1.**Read descriptive activity names from “activity\_labels.txt”**

activityLabels <- read.table(file.path(path\_rf, "activity\_labels.txt"),header = FALSE)

1. **facorize Variale activity in the data frame Data using descriptive activity names**
2. **check**

head(Data$activity,30)

## [1] STANDING STANDING STANDING STANDING STANDING STANDING STANDING

## [8] STANDING STANDING STANDING STANDING STANDING STANDING STANDING

## [15] STANDING STANDING STANDING STANDING STANDING STANDING STANDING

## [22] STANDING STANDING STANDING STANDING STANDING STANDING SITTING

## [29] SITTING SITTING

## 6 Levels: WALKING WALKING\_UPSTAIRS WALKING\_DOWNSTAIRS ... LAYING

**Appropriately labels the data set with descriptive variable names**

In the former part, variables activity and subject and names of the activities have been labelled using descriptive names.In this part, Names of Feteatures will labelled using descriptive variable names.

* prefix t is replaced by time
* Acc is replaced by Accelerometer
* Gyro is replaced by Gyroscope
* prefix f is replaced by frequency
* Mag is replaced by Magnitude
* BodyBody is replaced by Body

names(Data)<-gsub("^t", "time", names(Data))

names(Data)<-gsub("^f", "frequency", names(Data))

names(Data)<-gsub("Acc", "Accelerometer", names(Data))

names(Data)<-gsub("Gyro", "Gyroscope", names(Data))

names(Data)<-gsub("Mag", "Magnitude", names(Data))

names(Data)<-gsub("BodyBody", "Body", names(Data))

**Creates a second,independent tidy data set and ouput it**

In this part,a second, independent tidy data set will be created with the average of each variable for each activity and each subject based on the data set in step 4.

**library**(plyr);

Data2<-aggregate(. ~subject + activity, Data, mean)

Data2<-Data2[order(Data2$subject,Data2$activity),]

write.table(Data2, file = "tidydata.txt",row.name=FALSE)