Final Assignemnt: Building a tidy data Set.

Please note that this code requires dplyr and tidyr packages and will attempt to install them if they are missing.

```
if(!require(dplyr)){
    install.packages("dplyr")
}

if(!require(tidyr)){
    install.packages("tidyr")
}

library(dplyr)

library(tidyr)
```

Next, we shall download and read the data

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

download.file(URL, "UCI.zip", method = "libcurl")

unzip("UCI.zip")

test.raw <- read.table("UCI HAR Dataset/test/X_test.txt")

test.subject<- read.table("UCI HAR Dataset/test/subject_test.txt")

test.activity<- read.table("UCI HAR Dataset/test/Y_test.txt")

train.raw <- read.table("UCI HAR Dataset/train/X_train.txt")

train.subject<- read.table("UCI HAR Dataset/train/subject_train.txt")

train.activity<- read.table("UCI HAR Dataset/train/Y_train.txt")

activity.labels<- read.table("UCI HAR Dataset/activity_labels.txt")
```

Lets do a vertical merge for the test and train data, then add variable names. To make variable names more user friendly, lets decapitalize them and remove the “-” sign.

```
var.names<- read.table("UCI HAR Dataset/features.txt")

data.raw<-rbind(test.raw, train.raw)

names(data.raw)<-tolower(var.names[,2])

names(data.raw)<-gsub("-", " ", names(data.raw))

str(data.raw, list.len=15)
```

```
## 'data.frame':    10299 obs. of  561 variables:
##  $ tbodyacc mean() x                  : num  0.257 0.286 0.275 0.27 0.275 ...
##  $ tbodyacc mean() y                  : num  -0.0233 -0.0132 -0.0261 -0.0326 -0.0278 ...
##  $ tbodyacc mean() z                  : num  -0.0147 -0.1191 -0.1182 -0.1175 -0.1295 ...
##  $ tbodyacc std() x                   : num  -0.938 -0.975 -0.994 -0.995 -0.994 ...
##  $ tbodyacc std() y                   : num  -0.92 -0.967 -0.97 -0.973 -0.967 ...
##  $ tbodyacc std() z                   : num  -0.668 -0.945 -0.963 -0.967 -0.978 ...
##  $ tbodyacc mad() x                   : num  -0.953 -0.987 -0.994 -0.995 -0.994 ...
##  $ tbodyacc mad() y                   : num  -0.925 -0.968 -0.971 -0.974 -0.966 ...
##  $ tbodyacc mad() z                   : num  -0.674 -0.946 -0.963 -0.969 -0.977 ...
##  $ tbodyacc max() x                   : num  -0.894 -0.894 -0.939 -0.939 -0.939 ...
##  $ tbodyacc max() y                   : num  -0.555 -0.555 -0.569 -0.569 -0.561 ...
##  $ tbodyacc max() z                   : num  -0.466 -0.806 -0.799 -0.799 -0.826 ...
##  $ tbodyacc min() x                   : num  0.717 0.768 0.848 0.848 0.849 ...
##  $ tbodyacc min() y                   : num  0.636 0.684 0.668 0.668 0.671 ...
##  $ tbodyacc min() z                   : num  0.789 0.797 0.822 0.822 0.83 ...
##   [list output truncated]
```

Now, we shall select all variables names which contain the words “mean” and “std” (standard deviation). while excluding names with “freq” and “angle”.

```
list1<-grepl("mean", names(data.raw))

list2<-grepl("std", names(data.raw))

list3<-grepl("freq", names(data.raw))

list4<-grepl("angle", names(data.raw))

filter.list<- (list1 | list2) & (!list3 & !list4)

data.raw1<-data.raw[,filter.list]

str(data.raw, list.len=10)
```

```
## 'data.frame':    10299 obs. of  561 variables:
##  $ tbodyacc mean() x                  : num  0.257 0.286 0.275 0.27 0.275 ...
##  $ tbodyacc mean() y                  : num  -0.0233 -0.0132 -0.0261 -0.0326 -0.0278 ...
##  $ tbodyacc mean() z                  : num  -0.0147 -0.1191 -0.1182 -0.1175 -0.1295 ...
##  $ tbodyacc std() x                   : num  -0.938 -0.975 -0.994 -0.995 -0.994 ...
##  $ tbodyacc std() y                   : num  -0.92 -0.967 -0.97 -0.973 -0.967 ...
##  $ tbodyacc std() z                   : num  -0.668 -0.945 -0.963 -0.967 -0.978 ...
##  $ tbodyacc mad() x                   : num  -0.953 -0.987 -0.994 -0.995 -0.994 ...
##  $ tbodyacc mad() y                   : num  -0.925 -0.968 -0.971 -0.974 -0.966 ...
##  $ tbodyacc mad() z                   : num  -0.674 -0.946 -0.963 -0.969 -0.977 ...
##  $ tbodyacc max() x                   : num  -0.894 -0.894 -0.939 -0.939 -0.939 ...
##   [list output truncated]
```

Now that we have all selected the necessary variables, properly labelled them and combined all the readings into a single data frame, we need to add the subject and activity data.

Firstly, lets we shall column bind the test subject and test activity data, as well as train subject and train activity data. Then we row bind the resulting data frames and get the total subject and activity data.

```
subject.activity<- rbind(cbind(test.subject, test.activity), cbind(train.subject, train.activ
ity))
```

Currently, the activity type is recorded as integers. Lets assign some descriptive variable names to them.

```
subject.activity[,2]<- as.factor(subject.activity[,2])

levels(subject.activity[,2])<- c("walking", "walking upstairs", "walking downstairs", "sittin
g", "standing", "laying")

names(subject.activity)<-c("subject", "activity")

str(data.raw, list.len=10)
```

```
## 'data.frame':    10299 obs. of  561 variables:
##  $ tbodyacc mean() x                  : num  0.257 0.286 0.275 0.27 0.275 ...
##  $ tbodyacc mean() y                  : num  -0.0233 -0.0132 -0.0261 -0.0326 -0.0278 ...
##  $ tbodyacc mean() z                  : num  -0.0147 -0.1191 -0.1182 -0.1175 -0.1295 ...
##  $ tbodyacc std() x                   : num  -0.938 -0.975 -0.994 -0.995 -0.994 ...
##  $ tbodyacc std() y                   : num  -0.92 -0.967 -0.97 -0.973 -0.967 ...
##  $ tbodyacc std() z                   : num  -0.668 -0.945 -0.963 -0.967 -0.978 ...
##  $ tbodyacc mad() x                   : num  -0.953 -0.987 -0.994 -0.995 -0.994 ...
##  $ tbodyacc mad() y                   : num  -0.925 -0.968 -0.971 -0.974 -0.966 ...
##  $ tbodyacc mad() z                   : num  -0.674 -0.946 -0.963 -0.969 -0.977 ...
##  $ tbodyacc max() x                   : num  -0.894 -0.894 -0.939 -0.939 -0.939 ...
##   [list output truncated]
```

Next, lets combine the subject, activity and test data into a single data frame.

```
data.raw2<- cbind(subject.activity, data.raw1)

str(data.raw2, list.len=10)
```

```
## 'data.frame':    10299 obs. of  68 variables:
##  $ subject                    : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ activity                   : Factor w/ 6 levels "walking","walking upstairs",..: 5 5 5
 5 5 5 5 5 5 5 ...
##  $ tbodyacc mean() x          : num  0.257 0.286 0.275 0.27 0.275 ...
##  $ tbodyacc mean() y          : num  -0.0233 -0.0132 -0.0261 -0.0326 -0.0278 ...
##  $ tbodyacc mean() z          : num  -0.0147 -0.1191 -0.1182 -0.1175 -0.1295 ...
##  $ tbodyacc std() x           : num  -0.938 -0.975 -0.994 -0.995 -0.994 ...
##  $ tbodyacc std() y           : num  -0.92 -0.967 -0.97 -0.973 -0.967 ...
##  $ tbodyacc std() z           : num  -0.668 -0.945 -0.963 -0.967 -0.978 ...
##  $ tgravityacc mean() x       : num  0.936 0.927 0.93 0.929 0.927 ...
##  $ tgravityacc mean() y       : num  -0.283 -0.289 -0.288 -0.293 -0.303 ...
##   [list output truncated]
```

As you can see from the table above, each variable consists of 3 component variables, separated by the “-”. This goes against tidy data principles, hence we should decompose these variables accordingly. Also, for we shall remove the “()” next to “mean”, for aesthetic purposes

```
data.raw2 %>%

    gather(key = variable, value = result, -subject, -activity, factor_key = TRUE) %>%

    arrange(subject, activity) %>%

    separate(variable, c("variable", "statistic", "vector"), sep= " ")->tidy
```

```
## Warning: Too few values at 185382 locations: 2851, 2852, 2853, 2854, 2855,
## 2856, 2857, 2858, 2859, 2860, 2861, 2862, 2863, 2864, 2865, 2866, 2867,
## 2868, 2869, 2870, ...
```

```
tidy$statistic<-gsub('.{2}$', '',tidy$statistic)

head(tidy)
```

```
##   subject activity variable statistic vector    result
## 1       1  walking tbodyacc      mean      x 0.2820216
## 2       1  walking tbodyacc      mean      x 0.2558408
## 3       1  walking tbodyacc      mean      x 0.2548672
## 4       1  walking tbodyacc      mean      x 0.3433705
## 5       1  walking tbodyacc      mean      x 0.2762397
## 6       1  walking tbodyacc      mean      x 0.2554682
```

Lastly, we shall group our data, take the mean value of each variable and save it as a “Tidy Data.txt”, as per the requirements of this assignment.

```
tidy<-group_by(tidy, subject, activity, variable, statistic, vector)

final<- summarise(tidy, meanvalue=mean(result))

write.table(final, file="Tidy Data.txt", row.names = FALSE)

head(final)
```

```
## Source: local data frame [6 x 6]
## Groups: subject, activity, variable, statistic [2]
##
##   subject activity variable statistic vector   meanvalue
##     <int>   <fctr>    <chr>     <chr>  <chr>       <dbl>
## 1       1  walking fbodyacc      mean      x -0.20279431
## 2       1  walking fbodyacc      mean      y  0.08971273
## 3       1  walking fbodyacc      mean      z -0.33156012
## 4       1  walking fbodyacc       std      x -0.31913472
## 5       1  walking fbodyacc       std      y  0.05604001
## 6       1  walking fbodyacc       std      z -0.27968675
```