```
library(dplyr)
library(plyr)
```

1. Merging the training and the test sets to create one data set

(a) Extracted variable names from "features.txt" file which are used in naming columns in train and test set.

```
name <- read.table("features.txt")
test_set <- read.table("./test/X_test.txt", col.names = name$V2)
train_set <- read.table("./train/X_train.txt", col.names = name$V2)
```

(b) Extracted id and activity details of both the test and training set

```
test_id <- read.table("./test/subject_test.txt", col.names = "id")
test_label <- read.table("./test/y_test.txt", col.names = "activity")
train_id <- read.table("./train/subject_train.txt", col.names="id")
train_label <- read.table("./train/y_train.txt",col.names = "activity")
```

test_id and train_id has values of ids of test and train subjects. Their activities are stored in test_label and train_label respectively.

(c) Combined activity and set values with id of subjects

```
test <- cbind(test_id, test_label, test_set)
train <- cbind(train_id,train_label,train_set)
```

columns of id, activity, and set of test and train are combined individually.

(d) Merged both test and train into one data named as one_data

```
one_data <- rbind(test,train)
```

#2. Extracts only the measurements on the mean and standard deviation for each measurement.# (a) Variable names have mean() and std() for mean and standard deviation. (b) While extracting these columns, variables which have meanFreq are also selected. Therefore, in the next step they are deselected.

```
selected_data <- one_data %>%
    select(id, activity, contains(c("mean","std"), ignore.case = FALSE)) %>%
    select(-contains("meanFreq"))
```

3. Uses descriptive activity names to name the activities in the data set

(a) Activity of all subjects are changed as (Walk, Walk_up, Walk_down, Sit, Stand, Lay) from numbers 1 to 6, respectively.
(b) Data are also sorted with subjects and their activities.

```
selected_data <- selected_data %>%
  mutate(activity = revalue(as.character(activity),
c("1"="WALK","2"="WALK_UP","3"="WALK_DOWN", "4"= "SIT", "5" = "STAND", "6"= "LAY"))) %>%
  arrange(id, activity)
```

4. Appropriately labels the data set with descriptive variable names

The descriptive variable names are developed with the same way as explained in detail for the example : "tBodyAcc.mean...X (a) The variable names have dot(.) which are replaced by (-). It leads to"tBodayACC-mean-X" (b) The first word tBodyAcc is difficult to remember and there it is separated by dash as "t-Body-Acc-mean–X" (c) Double dash is removed to create "t-Body-Acc-mean-X"

```
name_var <- strsplit(names(selected_data), "\\.") %>%
        sapply(function(a) {a <- a[!a ==""];  a <- paste(a, collapse = "-")})
name_var <- strsplit(name_var, "(?<=.)(?=[[:upper:]])", perl=TRUE) %>%
                sapply(function(x) x <- paste(x, collapse = "-"))
name_var <- strsplit(name_var, "--") %>%
        sapply(function(a) {a <- paste(a, collapse = "-")})

names(selected_data) <- name_var
```

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

(a) Data is grouped by id and activity by using function group_by
(b) Subsequently, mean is calculated for each subject and each activity.
(c) Data is written in txt file.

```
mean_activity <- selected_data %>% group_by(id, activity) %>%
        summarise_all(~ mean(.x, na.rm = TRUE))
write.table(mean_activity, file = "Activity_mean.txt",row.name=FALSE)
```