Introduction

The variables used in this project are described in the files

a) Activity_labels.txt
b) Feature_info.txt
c) Features.txt

The subjects are assigned numbers from 1 to 30.

ACTVITY LABELS (from activity labels.txt)

```
1 WALKING
2 WALKING_UPSTAIRS
3 WALKING_DOWNSTAIRS
4 SITTING
5 STANDING
6 LAYING
```

ASSIGNED VARIABLES

There are three assigned (i.e. not measured or calculated)

a) subject.id - a numeric code used to identify the human subject (1 to 30)
b) activity.id – a numeric code used to identify the activity (1 to 6)
c) activity.name – a string describing the activity.id e.g. WALKING

MEASURED VARIABLES (FEATURES)

There are 561 measured or calculated variables which take on real values in the range -1 to +1. The format of these variables as given in the source data is not suitable for use in an R script because they contain characters that are illegal in variable names e.g. hyphens (-), commas (,) and left/right parentheses. The original feature names are renamed in the script as follows

a) Adjacent pairs of parentheses i.e. '()' are removed
b) Embedded parentheses are replaced by an underscore (_)
c) Parentheses appearing at the end of a variable are removed
d) Hyphens (-) and commas (,) are replaced by underscores (_) wherever they appear

The HTML file '**feature_name_transformation_table.html**' (as well as the CSV file **feature_name_transformation_table.csv)** shows each feature's original and transformed names. The first few are shown below. The transformation was chosen to make it easy to relate the new names to the descriptions given in the file **feature_info.txt.**

feature	original_feature_name	new_feature_name
1	tBodyAcc-mean()-X	tBodyAcc_mean_X
2	tBodyAcc-mean()-Y	tBodyAcc_mean_Y
3	tBodyAcc-mean()-Z	tBodyAcc_mean_Z
4	tBodyAcc-std()-X	tBodyAcc_std_X
5	tBodyAcc-std()-Y	tBodyAcc_std_Y
6	tBodyAcc-std()-Z	tBodyAcc_std_Z
7	tBodyAcc-mad()-X	tBodyAcc_mad_X
8	tBodyAcc-mad()-Y	tBodyAcc_mad_Y

The data sets requested in the assignment use all three assigned variables as well 48 of the measured / calculated variables. These variables are listed in the file ‘**variable_names.html**’ (as well as the CSV file **variable_names.csv**) along with their data types and allowed ranges. The first few values of the list are shown below.

Variable Name	Data Type	Range	Description
subject.id	Integer	1 to 30	Numeric code used to identify the subject
activity.id	Integer	1 to 6	Numeric code used to identify the activity
activity.name	Character		Alphanumeric string used to describe the activity e.g. WALKING
tBodyAcc_mean_X	Double	-1 to +1	Measured value of feature
tBodyAcc_mean_Y	Double	-1 to +1	Measured value of feature
tBodyAcc_mean_Z	Double	-1 to +1	Measured value of feature
tBodyAcc_std_X	Double	-1 to +1	Measured value of feature
tBodyAcc_std_Y	Double	-1 to +1	Measured value of feature

Processing

After extracting the zipped file contains into the working directory and sourcing the file ‘run_analysis.R’, the command **run_analysis()** starts the process.

The main function uses the worker function

get_data_set(directory=”./test”,data_set= ”test”)

to do the bulk of the processing. The worker function loads the five data sets

a) activity labels – from activity_labels.txt
b) features – from features.txt
c) subjects – from subject_test.txt or subject_train.txt depending on the paramneter **data_set**
d) subject activity list – from y_test.txt or y_train.txt depending on **data_set**
e) measurments – from x_test.txt or x_train.txt

After performing any necessary name transformations, data sets a) and d) are merged to create a labelled activity list (activity.id, activity.name).

Then the data sets e) and the labelled activity data set are merged to produce a labelled subject activity data set (activity.id, activity.name, all 561 features). We then merge the last data set to the subjects to produce the labelled subject activity listing with all 561 measurements which is returned to the main function.

The main function merges the two data sets (TEST and TRAIN) and then extracts all rows for the subset of columns {subject.id, activuity.id, acitivity.name and the mean and standard deviation columns) and writes this out to the CSV file **subject_activity_mean_std.csv** with row names supressed. The function then produces a summary data set grouped by (subject.id, activity.id, activity.name) and the mean of the measured variables. The summary data ser is written to the file **subject_activity_summary_mean_std.csv**.

The summarisation requires the package **dplyr** – it loads it if it’s not already in the environment. The function also writes out the column names of either data set to the file variable_names.csv to kick-start the production of part of the code book.