GCDProject

Getting and Cleaning Data Class Project Script

All the code for this project is contained in a single R script file called 'GCDProject.R'

The code is structured in a conventional 'C' language style, with some global variables declared first, then some functions that perform simple individual tasks, and final main body part that glues all the logic together. The global vars declared at the begining are URL and path names, in order to keep code tidy and readable, not cluttered with long path names embedded in the code, but also to make it more portable.

Step 0

The function getBaseData checks if we already have the original data to work with. If we don't, it downloads the ZIP file and expands it in the working directory. Then we read all individual data tables. Numeric data comes in fixed width format files that we read with the function readNumData. This function reads both test and train data sets (read code_book.md) and stacks them up in a single table data frame, making use of the readr package for faster reading. Similarly, readIdData is used to read index files, subject and activity code for each vector of features of the test and train data sets, and stack them up. Finally, the feature names read are not valid variable names (contain invalid symbols), and have to be made valid (make.names) and unique. The feature names list is pre-pended with "Subject" and "Activity" to provide variable names for those columns.
At this moment we have read and stored subject ids (subject), activity ids (activity) and names (activity_names), feature vectors (features) and names (feature_names). {r eval=FALSE} downloadBaseData()
activity_names <- read.table(activity_labels_f, row.names = 1, colClasses = c("integer","character"))
feature_names <- read.table(features_f, row.names = 1, colClasses = c("integer","character"))
features   <- readNumData(X_test_f)   activity   <- readIdData(y_test_f)   subject
<- readIdData(subject_test_f)   feature_names <- c("Subject","Activity",make.names(feature_names[,1],u
= TRUE)) # Valid col names ###Step 1### Subject ids, activity ids, and feature vectors are all column bound to create a single table, to which we asign the feature names as column names. {r eval=FALSE} baseData   <- cbind(subject,activity,features) # Combine columns
colnames(baseData) <- feature_names   # and column names