

# Code Book: Getting and Cleaning Data Course Project

Ron Cox

8/13/2021

## Introduction

This code book describes the variables, the data, and any transformations or work that I performed to clean up the data related to the project in order to develop an R script called *run\_analysis.R* that meets the following requirements.

R1: Merges the training and the test sets to create one data set. See Step E: Merge Train/Test Sets

R2: Extracts only the measurements on the mean and standard deviation for each measurement. See Step C: Build Train/Test Data Sets

R3: Uses descriptive activity names to name the activities in the data set. See Step D: Append Activity Labels

R4: Appropriately labels the data set with descriptive variable names. See Step C: Build Train/Test Data Sets

R5: 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. See Step F: Average Variables by Activity and Subject

## Step-By-Step Procedure

### Step A: Download compressed data set, retrieve select files, and load them into the environment

This step is the prerequisite of the project. The zipfile is downloaded to a temporary file and the folder structure is stored in the **zipFiles** variable. After identifying the specific files required for the project, those files were read directly from the compressed temporary file into the following data frames using the indexes in the **zipFiles** variable:

Object Names	File Names	Full Path (in zip file)
activityLabels	activity_labels.txt	[1] "UCI HAR Dataset/activity_labels.txt"
features	features.txt	[2] "UCI HAR Dataset/features.txt"
testSubjects	subject_test.txt	[16] "UCI HAR Dataset/test/subject_test.txt"
testSet	X_test.txt	[17] "UCI HAR Dataset/test/X_test.txt"
activityLabels_test	y_test.txt	[18] "UCI HAR Dataset/test/y_test.txt"
trainSubjects	subject_train.txt	[30] "UCI HAR Dataset/train/subject_train.txt"
trainSet	X_train.txt	[31] "UCI HAR Dataset/train/X_train.txt"
activityLabels_train	y_train.txt	[32] "UCI HAR Dataset/train/y_train.txt"

```

# Create a temporary file to store the downloaded zip file
fileUrl <- "https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"
temp <- tempfile()
download.file(fileUrl, temp)

# Get a list of the files in the zip file
zipFiles <- unzip(zipfile = temp, list = TRUE)

## The following files will be loaded into the project using the corresponding index
## of zipFiles:
##          1 = activity_labels.txt
##          2 = features.txt
##         16 = subject_test.txt
##         17 = X_test.txt
##         18 = y_test.txt
##        30 = subject_train.txt
##        31 = X_train.txt
##        32 = y_train.txt

# Read in activity_labels.txt
activityLabels <- read.csv(unz(temp, zipFiles$Name[1]),
                          header = FALSE, sep = "",
                          col.names = c("ActivityCode", "Activity"))

# Read in features.txt
features <- read.csv(unz(temp, zipFiles$Name[2]),
                    header = FALSE, sep = "",
                    col.names = c("FeatureCode", "Feature"))

# Read each of the three data sets related to testing: subject_test.txt, X_test.txt,
# and y_test.txt
testSubjects <- read.csv(unz(temp, zipFiles$Name[16]),
                        header = FALSE, col.names = "Subjects")

testSet <- read.csv(unz(temp, zipFiles$Name[17]),
                  header = FALSE, sep = "") #Use "features" for column names

activityLabels_test <- read.csv(unz(temp, zipFiles$Name[18]),
                              header = FALSE, col.names = "Activity")

# Read each of the three data sets related to training: subject_train.txt, X_train.txt,
# and y_train.txt
trainSubjects <- read.csv(unz(temp, zipFiles$Name[30]),
                        header = FALSE, col.names = "Subjects")

trainSet <- read.csv(unz(temp, zipFiles$Name[31]),
                  header = FALSE, sep = "") #Use "features" for column names

activityLabels_train <- read.csv(unz(temp, zipFiles$Name[32]),
                              header = FALSE, col.names = "Activity")

# Close the connection to the temporary file
unlink(temp)

```

## Step B: Data Exploration

I reviewed each data frame to gain a better understanding of the data sets and to determine how the original files were related to each other.

### activityLabels

```
# Get class
class(activityLabels)

## [1] "data.frame"

# Get dimension
dim(activityLabels)

## [1] 6 2

# Get structure
str(activityLabels)

## 'data.frame': 6 obs. of 2 variables:
## $ ActivityCode: int 1 2 3 4 5 6
## $ Activity : chr "WALKING" "WALKING_UPSTAIRS" "WALKING_DOWNSTAIRS" "SITTING" ...

# Display values
activityLabels

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

### features

```
# Get class
class(features)

## [1] "data.frame"

# Get dimension
dim(features)

## [1] 561 2
```

```
# Get structure
str(features)
```

```
## 'data.frame': 561 obs. of 2 variables:
## $ FeatureCode: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Feature : chr "tBodyAcc-mean()-X" "tBodyAcc-mean()-Y" "tBodyAcc-mean()-Z" "tBodyAcc-std()-X"
```

```
# Display values
head(features)
```

```
## FeatureCode Feature
## 1 1 tBodyAcc-mean()-X
## 2 2 tBodyAcc-mean()-Y
## 3 3 tBodyAcc-mean()-Z
## 4 4 tBodyAcc-std()-X
## 5 5 tBodyAcc-std()-Y
## 6 6 tBodyAcc-std()-Z
```

testSubjects

```
# Get class
class(testSubjects)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(testSubjects)
```

```
## [1] 2947 1
```

```
# Get structure
str(testSubjects)
```

```
## 'data.frame': 2947 obs. of 1 variable:
## $ Subjects: int 2 2 2 2 2 2 2 2 2 2 ...
```

```
# Display values
head(testSubjects)
```

```
## Subjects
## 1 2
## 2 2
## 3 2
## 4 2
## 5 2
## 6 2
```

testSet

```
# Get class
class(testSet)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(testSet)
```

```
## [1] 2947 561
```

```
# Get structure (limited to first 10 for brevity)
str(testSet, list.len = 10)
```

```
## '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 ...
## [list output truncated]
```

```
# Display values (limited to a 5 row, 5 column set for brevity)
testSet[1:5,1:5]
```

```
##           V1           V2           V3           V4           V5
## 1 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908
## 2 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579
## 3 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255
## 4 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676
## 5 0.2748330 -0.02784779 -0.12952716 -0.9938525 -0.9674455
```

activityLabels\_test

```
# Get class
class(activityLabels_test)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(activityLabels_test)
```

```
## [1] 2947 1
```

```
# Get structure
str(activityLabels_test)
```

```
## 'data.frame': 2947 obs. of 1 variable:
## $ Activity: int 5 5 5 5 5 5 5 5 5 5 ...
```

```
# Display values
head(activityLabels_test)
```

```
## Activity
## 1      5
## 2      5
## 3      5
## 4      5
## 5      5
## 6      5
```

**trainSubjects**

```
# Get class
class(trainSubjects)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(trainSubjects)
```

```
## [1] 7352 1
```

```
# Get structure
str(trainSubjects)
```

```
## 'data.frame': 7352 obs. of 1 variable:
## $ Subjects: int 1 1 1 1 1 1 1 1 1 1 ...
```

```
# Display values
head(trainSubjects)
```

```
## Subjects
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

**trainSet**

```
# Get class
class(trainSet)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(trainSet)
```

```
## [1] 7352 561
```

```
# Get structure (limited to first 10 for brevity)
str(trainSet, list.len = 10)
```

```
## '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 ...
## [list output truncated]
```

```
# Display values (limited to a 5 row, 5 column set for brevity)
trainSet[1:5,1:5]
```

```
##           V1           V2           V3           V4           V5
## 1 0.2885845 -0.02029417 -0.1329051 -0.9952786 -0.9831106
## 2 0.2784188 -0.01641057 -0.1235202 -0.9982453 -0.9753002
## 3 0.2796531 -0.01946716 -0.1134617 -0.9953796 -0.9671870
## 4 0.2791739 -0.02620065 -0.1232826 -0.9960915 -0.9834027
## 5 0.2766288 -0.01656965 -0.1153619 -0.9981386 -0.9808173
```

activityLabels\_train

```
# Get class
class(activityLabels_train)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(activityLabels_train)
```

```
## [1] 7352 1
```

```
# Get structure
str(activityLabels_train)
```

```
## 'data.frame': 7352 obs. of 1 variable:
## $ Activity: int 5 5 5 5 5 5 5 5 5 5 ...
```

```
# Display values
head(activityLabels_train)
```

```
## Activity
## 1      5
## 2      5
## 3      5
## 4      5
## 5      5
## 6      5
```

## Step C: Build Train/Test Data Sets

NOTE: Because the steps were exactly the same for both the train and test data sets, I performed the sequences together rather than assembling one set and then another.

This step meets R2: Extracts only the measurements on the mean and standard deviation for each measurement.

This step meets R4: Appropriately labels the data set with descriptive variable names.

```
# Use the values from the features data set to establish column names for testSet and trainSet
names(testSet) <- features$Feature
names(trainSet) <- features$Feature

# Identify all values in features data set that contains the string 'mean' or 'std'
targetedFeatures <- features %>% filter(grepl("mean|std", Feature))

# Filter out all columns in testSet and trainSet where the variable name does not include
# the string 'mean' or 'std'
testSet <- testSet[,targetedFeatures$FeatureCode]
trainSet <- trainSet[,targetedFeatures$FeatureCode]
```

trainSet

```
# Get class
class(trainSet)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(trainSet)
```

```
## [1] 7352 79
```



```
# Get structure (limited to first 10 for brevity)
str(trainSet, list.len = 10)
```

```
## 'data.frame':    7352 obs. of  79 variables:
## $ tBodyAcc-mean()-X      : num  0.289 0.278 0.28 0.279 0.277 ...
## $ tBodyAcc-mean()-Y      : num  -0.0203 -0.0164 -0.0195 -0.0262 -0.0166 ...
## $ tBodyAcc-mean()-Z      : num  -0.133 -0.124 -0.113 -0.123 -0.115 ...
## $ tBodyAcc-std()-X       : num  -0.995 -0.998 -0.995 -0.996 -0.998 ...
## $ tBodyAcc-std()-Y       : num  -0.983 -0.975 -0.967 -0.983 -0.981 ...
## $ tBodyAcc-std()-Z       : num  -0.914 -0.96 -0.979 -0.991 -0.99 ...
## $ tGravityAcc-mean()-X   : num  0.963 0.967 0.967 0.968 0.968 ...
## $ tGravityAcc-mean()-Y   : num  -0.141 -0.142 -0.142 -0.144 -0.149 ...
## $ tGravityAcc-mean()-Z   : num  0.1154 0.1094 0.1019 0.0999 0.0945 ...
## $ tGravityAcc-std()-X    : num  -0.985 -0.997 -1 -0.997 -0.998 ...
## [list output truncated]
```

```
# Display values (limited to a 5 row, 5 column set for brevity)
trainSet[1:5,1:5]
```

```
##      tBodyAcc-mean()-X tBodyAcc-mean()-Y tBodyAcc-mean()-Z tBodyAcc-std()-X
## 1      0.2885845      -0.02029417      -0.1329051      -0.9952786
## 2      0.2784188      -0.01641057      -0.1235202      -0.9982453
## 3      0.2796531      -0.01946716      -0.1134617      -0.9953796
## 4      0.2791739      -0.02620065      -0.1232826      -0.9960915
## 5      0.2766288      -0.01656965      -0.1153619      -0.9981386
##      tBodyAcc-std()-Y
## 1      -0.9831106
## 2      -0.9753002
## 3      -0.9671870
## 4      -0.9834027
## 5      -0.9808173
```

```
testSet
```

```
# Get class
class(testSet)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(testSet)
```

```
## [1] 2947    79
```

```
# Get structure (limited to first 10 for brevity)
str(testSet, list.len = 10)
```

```
## 'data.frame':    2947 obs. of  79 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 ...
## $ 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 ...
## $ tGravityAcc-mean()-Z   : num   0.115 0.153 0.146 0.143 0.138 ...
## $ tGravityAcc-std()-X    : num  -0.925 -0.989 -0.996 -0.993 -0.996 ...
## [list output truncated]
```

```
# Display values (limited to a 5 row, 5 column set for brevity)
testSet[1:5,1:5]
```

```
##      tBodyAcc-mean()-X tBodyAcc-mean()-Y tBodyAcc-mean()-Z tBodyAcc-std()-X
## 1      0.2571778      -0.02328523      -0.01465376      -0.9384040
## 2      0.2860267      -0.01316336      -0.11908252      -0.9754147
## 3      0.2754848      -0.02605042      -0.11815167      -0.9938190
## 4      0.2702982      -0.03261387      -0.11752018      -0.9947428
## 5      0.2748330      -0.02784779      -0.12952716      -0.9938525
##      tBodyAcc-std()-Y
## 1      -0.9200908
## 2      -0.9674579
## 3      -0.9699255
## 4      -0.9732676
## 5      -0.9674455
```

## Step D: Append Activity Labels

This step meets R3: Uses descriptive activity names to name the activities in the data set.

```
# Add activity labels to the test and train data sets
testActivities <- merge(activityLabels, activityLabels_test, by.x = "ActivityCode",
                        by.y = "Activity", all.y = TRUE)
trainActivities <- merge(activityLabels, activityLabels_train, by.x = "ActivityCode",
                        by.y = "Activity", all.y = TRUE)

# Add new variable, dataType, to test and train data sets
testActivities <- mutate(testActivities, DataType = "Test")
trainActivities <- mutate(trainActivities, DataType = "Train")
```

```
trainActivities
```

```
# Get class
class(trainActivities)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(trainActivities)
```

```
## [1] 7352    3
```

```
# Get structure
str(trainActivities)
```

```
## 'data.frame':    7352 obs. of  3 variables:
## $ ActivityCode: int  1 1 1 1 1 1 1 1 1 1 ...
## $ Activity    : chr  "WALKING" "WALKING" "WALKING" "WALKING" ...
## $ DataType    : chr  "Train" "Train" "Train" "Train" ...
```

```
# Display values
head(trainActivities)
```

```
##   ActivityCode Activity DataType
## 1             1  WALKING    Train
## 2             1  WALKING    Train
## 3             1  WALKING    Train
## 4             1  WALKING    Train
## 5             1  WALKING    Train
## 6             1  WALKING    Train
```

testActivities

```
# Get class
class(testActivities)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(testActivities)
```

```
## [1] 2947    3
```

```
# Get structure
str(testActivities)
```

```
## 'data.frame':    2947 obs. of  3 variables:
## $ ActivityCode: int  1 1 1 1 1 1 1 1 1 1 ...
## $ Activity    : chr  "WALKING" "WALKING" "WALKING" "WALKING" ...
## $ DataType    : chr  "Test" "Test" "Test" "Test" ...
```

```
# Display values
head(testActivities)
```

```
##   ActivityCode Activity DataType
## 1             1   WALKING      Test
## 2             1   WALKING      Test
## 3             1   WALKING      Test
## 4             1   WALKING      Test
## 5             1   WALKING      Test
## 6             1   WALKING      Test
```

## Step E: Merge Train/Test Sets

This step meets R1: Merges the training and the test sets to create one data set.

```
# Combine the training related data sets into a unified data frame
trainingData <- cbind(trainSubjects, trainActivities, trainSet)

# Combine the test related data sets into a unified data frame
testData <- cbind(testSubjects, testActivities, testSet)

# Combine the training and test data sets into a single data frame
combinedData <- rbind(trainingData, testData)

# Remove the ActivityCode variable
combinedData <- select(combinedData, -ActivityCode)
```

trainingData

```
# Get class
class(trainingData)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(trainingData)
```

```
## [1] 7352   83
```

```
# Get structure (limited to first 10 for brevity)
str(trainingData, list.len = 10)
```

```
## 'data.frame':   7352 obs. of  83 variables:
##  $ Subjects          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ActivityCode      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Activity          : chr  "WALKING" "WALKING" "WALKING" "WALKING" ...
##  $ DataType          : chr  "Train" "Train" "Train" "Train" ...
##  $ tBodyAcc-mean()-X : num  0.289 0.278 0.28 0.279 0.277 ...
##  $ tBodyAcc-mean()-Y : num -0.0203 -0.0164 -0.0195 -0.0262 -0.0166 ...
##  $ tBodyAcc-mean()-Z : num -0.133 -0.124 -0.113 -0.123 -0.115 ...
##  $ tBodyAcc-std()-X  : num -0.995 -0.998 -0.995 -0.996 -0.998 ...
##  $ tBodyAcc-std()-Y  : num -0.983 -0.975 -0.967 -0.983 -0.981 ...
##  $ tBodyAcc-std()-Z  : num -0.914 -0.96 -0.979 -0.991 -0.99 ...
##  [list output truncated]
```

```
# Display values (limited to a 5 row, 6 column set for brevity)
trainingData[1:5,1:6]
```

```
##   Subjects ActivityCode Activity DataType tBodyAcc-mean()-X tBodyAcc-mean()-Y
## 1         1           1 WALKING   Train         0.2885845        -0.02029417
## 2         1           1 WALKING   Train         0.2784188        -0.01641057
## 3         1           1 WALKING   Train         0.2796531        -0.01946716
## 4         1           1 WALKING   Train         0.2791739        -0.02620065
## 5         1           1 WALKING   Train         0.2766288        -0.01656965
```

```
testData
```

```
# Get class
class(testData)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(testData)
```

```
## [1] 2947   83
```

```
# Get structure (limited to first 10 for brevity)
str(testData, list.len = 10)
```

```
## 'data.frame':   2947 obs. of  83 variables:
## $ Subjects          : int  2 2 2 2 2 2 2 2 2 2 ...
## $ ActivityCode      : int  1 1 1 1 1 1 1 1 1 1 ...
## $ Activity          : chr  "WALKING" "WALKING" "WALKING" "WALKING" ...
## $ DataType          : chr  "Test" "Test" "Test" "Test" ...
## $ 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 ...
## [list output truncated]
```

```
# Display values (limited to a 5 row, 6 column set for brevity)
testData[1:5,1:6]
```

```
##   Subjects ActivityCode Activity DataType tBodyAcc-mean()-X tBodyAcc-mean()-Y
## 1         2           1 WALKING   Test         0.2571778        -0.02328523
## 2         2           1 WALKING   Test         0.2860267        -0.01316336
## 3         2           1 WALKING   Test         0.2754848        -0.02605042
## 4         2           1 WALKING   Test         0.2702982        -0.03261387
## 5         2           1 WALKING   Test         0.2748330        -0.02784779
```

combinedData

```
# Get class
class(combinedData)
```

```
## [1] "data.frame"
```

```
# Get dimension
dim(combinedData)
```

```
## [1] 10299    82
```

```
# Get structure (limited to first 10 for brevity)
str(combinedData, list.len = 10)
```

```
## 'data.frame':    10299 obs. of  82 variables:
## $ Subjects          : int  1 1 1 1 1 1 1 1 1 1 ...
## $ Activity          : chr  "WALKING" "WALKING" "WALKING" "WALKING" ...
## $ DataType          : chr  "Train" "Train" "Train" "Train" ...
## $ tBodyAcc-mean()-X : num  0.289 0.278 0.28 0.279 0.277 ...
## $ tBodyAcc-mean()-Y : num  -0.0203 -0.0164 -0.0195 -0.0262 -0.0166 ...
## $ tBodyAcc-mean()-Z : num  -0.133 -0.124 -0.113 -0.123 -0.115 ...
## $ tBodyAcc-std()-X   : num  -0.995 -0.998 -0.995 -0.996 -0.998 ...
## $ tBodyAcc-std()-Y   : num  -0.983 -0.975 -0.967 -0.983 -0.981 ...
## $ tBodyAcc-std()-Z   : num  -0.914 -0.96 -0.979 -0.991 -0.99 ...
## $ tGravityAcc-mean()-X : num  0.963 0.967 0.967 0.968 0.968 ...
## [list output truncated]
```

```
# Display values (limited to a 5 row, 7 column set for brevity)
combinedData[1:5,1:7]
```

```
##   Subjects Activity DataType tBodyAcc-mean()-X tBodyAcc-mean()-Y
## 1         1  WALKING   Train         0.2885845        -0.02029417
## 2         1  WALKING   Train         0.2784188        -0.01641057
## 3         1  WALKING   Train         0.2796531        -0.01946716
## 4         1  WALKING   Train         0.2791739        -0.02620065
## 5         1  WALKING   Train         0.2766288        -0.01656965
##   tBodyAcc-mean()-Z tBodyAcc-std()-X
## 1        -0.1329051        -0.9952786
## 2        -0.1235202        -0.9982453
## 3        -0.1134617        -0.9953796
## 4        -0.1232826        -0.9960915
## 5        -0.1153619        -0.9981386
```

## Step F: Average Variables by Activity and Subject

This step meets R5: 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.

```
# Calculate the mean of all measurement rows by subject and activity
tidyData <- combinedData %>%
  select(-DataType) %>%
  group_by(Subjects,Activity) %>%
  summarise(
    across(.cols = everything(), mean, .names = "mean_{.col}"),
    .groups = "keep")

# Write out tidyData data set to TidyData.csv in working directory
write.table(tidyData, file = "./TidyData_v1.csv", sep = ",", row.names = FALSE)
```

**tidyData**

```
# Get class
class(tidyData)
```

```
## [1] "grouped_df" "tbl_df"      "tbl"        "data.frame"
```

```
# Get dimension
dim(tidyData)
```

```
## [1] 40 81
```

```
# Get structure
str(tidyData)
```

```
## grouped_df [40 x 81] (S3: grouped_df/tbl_df/tbl/data.frame)
## $ Subjects                : int [1:40] 1 2 3 4 4 5 6 6 7 8 ...
## $ Activity                : chr [1:40] "WALKING" "WALKING" "WALKING" "WALKING" ...
## $ mean_tBodyAcc-mean()-X  : num [1:40] 0.266 0.273 0.273 0.277 0.27 ...
## $ mean_tBodyAcc-mean()-Y  : num [1:40] -0.0183 -0.0191 -0.0179 -0.0133 -0.0171 ...
## $ mean_tBodyAcc-mean()-Z  : num [1:40] -0.108 -0.116 -0.106 -0.106 -0.11 ...
## $ mean_tBodyAcc-std()-X   : num [1:40] -0.546 -0.606 -0.623 -0.684 -0.481 ...
## $ mean_tBodyAcc-std()-Y   : num [1:40] -0.368 -0.429 -0.48 -0.59 -0.384 ...
## $ mean_tBodyAcc-std()-Z   : num [1:40] -0.503 -0.589 -0.654 -0.742 -0.658 ...
## $ mean_tGravityAcc-mean()-X : num [1:40] 0.745 0.661 0.708 0.731 0.667 ...
## $ mean_tGravityAcc-mean()-Y : num [1:40] -0.0826 -0.1472 -0.0261 0.0723 0.1761 ...
## $ mean_tGravityAcc-mean()-Z : num [1:40] 0.0723 0.1349 0.0481 0.194 0.117 ...
## $ mean_tGravityAcc-std()-X  : num [1:40] -0.96 -0.963 -0.966 -0.968 -0.955 ...
## $ mean_tGravityAcc-std()-Y  : num [1:40] -0.951 -0.96 -0.945 -0.942 -0.941 ...
## $ mean_tGravityAcc-std()-Z  : num [1:40] -0.926 -0.945 -0.927 -0.937 -0.949 ...
## $ mean_tBodyAccJerk-mean()-X : num [1:40] 0.0771 0.0785 0.0702 0.078 0.0802 ...
## $ mean_tBodyAccJerk-mean()-Y : num [1:40] 0.01659 0.00709 0.01447 -0.00103 0.01068 ...
## $ mean_tBodyAccJerk-mean()-Z : num [1:40] -0.009108 0.000756 -0.000527 -0.007371 -0.007379
## $ mean_tBodyAccJerk-std()-X  : num [1:40] -0.525 -0.558 -0.635 -0.701 -0.523 ...
## $ mean_tBodyAccJerk-std()-Y  : num [1:40] -0.47 -0.492 -0.557 -0.671 -0.447 ...
## $ mean_tBodyAccJerk-std()-Z  : num [1:40] -0.717 -0.742 -0.796 -0.86 -0.818 ...
## $ mean_tBodyGyro-mean()-X    : num [1:40] -0.0209 -0.0517 -0.0248 -0.0299 -0.0275 ...
## $ mean_tBodyGyro-mean()-Y    : num [1:40] -0.0881 -0.0568 -0.0744 -0.075 -0.0846 ...
## $ mean_tBodyGyro-mean()-Z    : num [1:40] 0.0863 0.0873 0.0867 0.0957 0.0958 ...
```

```

## $ mean_tBodyGyro-std()-X : num [1:40] -0.687 -0.711 -0.699 -0.751 -0.623 ...
## $ mean_tBodyGyro-std()-Y : num [1:40] -0.451 -0.723 -0.763 -0.829 -0.75 ...
## $ mean_tBodyGyro-std()-Z : num [1:40] -0.597 -0.635 -0.709 -0.746 -0.696 ...
## $ mean_tBodyGyroJerk-mean()-X : num [1:40] -0.0971 -0.0876 -0.0992 -0.1051 -0.1122 ...
## $ mean_tBodyGyroJerk-mean()-Y : num [1:40] -0.0417 -0.0434 -0.0402 -0.0411 -0.0384 ...
## $ mean_tBodyGyroJerk-mean()-Z : num [1:40] -0.0471 -0.0558 -0.0521 -0.0609 -0.053 ...
## $ mean_tBodyGyroJerk-std()-X : num [1:40] -0.638 -0.672 -0.689 -0.778 -0.638 ...
## $ mean_tBodyGyroJerk-std()-Y : num [1:40] -0.634 -0.784 -0.843 -0.911 -0.888 ...
## $ mean_tBodyGyroJerk-std()-Z : num [1:40] -0.665 -0.675 -0.743 -0.791 -0.682 ...
## $ mean_tBodyAccMag-mean() : num [1:40] -0.454 -0.535 -0.563 -0.638 -0.442 ...
## $ mean_tBodyAccMag-std() : num [1:40] -0.497 -0.553 -0.591 -0.67 -0.506 ...
## $ mean_tGravityAccMag-mean() : num [1:40] -0.454 -0.535 -0.563 -0.638 -0.442 ...
## $ mean_tGravityAccMag-std() : num [1:40] -0.497 -0.553 -0.591 -0.67 -0.506 ...
## $ mean_tBodyAccJerkMag-mean() : num [1:40] -0.545 -0.588 -0.65 -0.722 -0.552 ...
## $ mean_tBodyAccJerkMag-std() : num [1:40] -0.516 -0.512 -0.608 -0.71 -0.548 ...
## $ mean_tBodyGyroMag-mean() : num [1:40] -0.475 -0.615 -0.643 -0.708 -0.574 ...
## $ mean_tBodyGyroMag-std() : num [1:40] -0.5 -0.681 -0.674 -0.749 -0.64 ...
## $ mean_tBodyGyroJerkMag-mean() : num [1:40] -0.64 -0.747 -0.784 -0.85 -0.771 ...
## $ mean_tBodyGyroJerkMag-std() : num [1:40] -0.652 -0.74 -0.804 -0.871 -0.803 ...
## $ mean_fBodyAcc-mean()-X : num [1:40] -0.532 -0.574 -0.626 -0.699 -0.518 ...
## $ mean_fBodyAcc-mean()-Y : num [1:40] -0.406 -0.433 -0.502 -0.611 -0.392 ...
## $ mean_fBodyAcc-mean()-Z : num [1:40] -0.596 -0.63 -0.7 -0.785 -0.708 ...
## $ mean_fBodyAcc-std()-X : num [1:40] -0.553 -0.62 -0.624 -0.679 -0.469 ...
## $ mean_fBodyAcc-std()-Y : num [1:40] -0.39 -0.465 -0.503 -0.606 -0.422 ...
## $ mean_fBodyAcc-std()-Z : num [1:40] -0.499 -0.601 -0.657 -0.74 -0.66 ...
## $ mean_fBodyAcc-meanFreq()-X : num [1:40] -0.181 -0.107 -0.193 -0.156 -0.23 ...
## $ mean_fBodyAcc-meanFreq()-Y : num [1:40] 0.0576 0.1057 0.0389 0.052 0.1364 ...
## $ mean_fBodyAcc-meanFreq()-Z : num [1:40] 0.05837 0.08721 0.00447 -0.01384 -0.02046 ...
## $ mean_fBodyAccJerk-mean()-X : num [1:40] -0.547 -0.562 -0.646 -0.718 -0.548 ...
## $ mean_fBodyAccJerk-mean()-Y : num [1:40] -0.507 -0.509 -0.583 -0.685 -0.471 ...
## $ mean_fBodyAccJerk-mean()-Z : num [1:40] -0.695 -0.716 -0.78 -0.845 -0.793 ...
## $ mean_fBodyAccJerk-std()-X : num [1:40] -0.544 -0.595 -0.658 -0.711 -0.54 ...
## $ mean_fBodyAccJerk-std()-Y : num [1:40] -0.466 -0.509 -0.56 -0.679 -0.459 ...
## $ mean_fBodyAccJerk-std()-Z : num [1:40] -0.738 -0.767 -0.811 -0.872 -0.842 ...
## $ mean_fBodyAccJerk-meanFreq()-X : num [1:40] -0.0488 0.0868 -0.0477 0.0388 -0.0627 ...
## $ mean_fBodyAccJerk-meanFreq()-Y : num [1:40] -0.215 -0.14 -0.237 -0.174 -0.19 ...
## $ mean_fBodyAccJerk-meanFreq()-Z : num [1:40] -0.0964 -0.0943 -0.1937 -0.0765 -0.1643 ...
## $ mean_fBodyGyro-mean()-X : num [1:40] -0.623 -0.639 -0.642 -0.708 -0.537 ...
## $ mean_fBodyGyro-mean()-Y : num [1:40] -0.505 -0.722 -0.775 -0.852 -0.799 ...
## $ mean_fBodyGyro-mean()-Z : num [1:40] -0.554 -0.602 -0.671 -0.724 -0.635 ...
## $ mean_fBodyGyro-std()-X : num [1:40] -0.708 -0.735 -0.719 -0.766 -0.652 ...
## $ mean_fBodyGyro-std()-Y : num [1:40] -0.43 -0.727 -0.759 -0.818 -0.727 ...
## $ mean_fBodyGyro-std()-Z : num [1:40] -0.65 -0.683 -0.751 -0.779 -0.749 ...
## $ mean_fBodyGyro-meanFreq()-X : num [1:40] -0.03203 -0.00577 -0.10257 -0.14535 -0.0762 ...
## $ mean_fBodyGyro-meanFreq()-Y : num [1:40] -0.0902 -0.0702 -0.1677 -0.3576 -0.4033 ...
## $ mean_fBodyGyro-meanFreq()-Z : num [1:40] -0.018 0.0514 -0.0322 -0.0236 0.0996 ...
## $ mean_fBodyAccMag-mean() : num [1:40] -0.478 -0.515 -0.579 -0.668 -0.498 ...
## $ mean_fBodyAccMag-std() : num [1:40] -0.59 -0.647 -0.663 -0.725 -0.591 ...
## $ mean_fBodyAccMag-meanFreq() : num [1:40] 0.1421 0.1922 0.0771 0.0692 0.1393 ...
## $ mean_fBodyBodyAccJerkMag-mean() : num [1:40] -0.499 -0.51 -0.605 -0.702 -0.529 ...
## $ mean_fBodyBodyAccJerkMag-std() : num [1:40] -0.542 -0.519 -0.616 -0.723 -0.577 ...
## $ mean_fBodyBodyAccJerkMag-meanFreq() : num [1:40] 0.2 0.193 0.112 0.165 0.153 ...
## $ mean_fBodyBodyGyroMag-mean() : num [1:40] -0.535 -0.7 -0.717 -0.785 -0.683 ...
## $ mean_fBodyBodyGyroMag-std() : num [1:40] -0.567 -0.725 -0.704 -0.769 -0.675 ...

```



```
## $ mean_fBodyBodyGyroMag-meanFreq() : num [1:40] 0.0649 0.0471 -0.0846 -0.1579 -0.0842 ...
## $ mean_fBodyBodyGyroJerkMag-mean() : num [1:40] -0.646 -0.752 -0.81 -0.868 -0.793 ...
## $ mean_fBodyBodyGyroJerkMag-std() : num [1:40] -0.686 -0.744 -0.81 -0.884 -0.83 ...
## $ mean_fBodyBodyGyroJerkMag-meanFreq(): num [1:40] 0.198 0.125 0.06 0.145 0.181 ...
## - attr(*, "groups")= tibble [40 x 3] (S3: tbl_df/tbl/data.frame)
## ..$ Subjects: int [1:40] 1 2 3 4 4 5 6 6 7 8 ...
## ..$ Activity: chr [1:40] "WALKING" "WALKING" "WALKING" "WALKING" ...
## ..$ .rows : list<int> [1:40]
## .. ..$ : int 1
## .. ..$ : int 2
## .. ..$ : int 3
## .. ..$ : int 4
## .. ..$ : int 5
## .. ..$ : int 6
## .. ..$ : int 7
## .. ..$ : int 8
## .. ..$ : int 9
## .. ..$ : int 10
## .. ..$ : int 11
## .. ..$ : int 12
## .. ..$ : int 13
## .. ..$ : int 14
## .. ..$ : int 15
## .. ..$ : int 16
## .. ..$ : int 17
## .. ..$ : int 18
## .. ..$ : int 19
## .. ..$ : int 20
## .. ..$ : int 21
## .. ..$ : int 22
## .. ..$ : int 23
## .. ..$ : int 24
## .. ..$ : int 25
## .. ..$ : int 26
## .. ..$ : int 27
## .. ..$ : int 28
## .. ..$ : int 29
## .. ..$ : int 30
## .. ..$ : int 31
## .. ..$ : int 32
## .. ..$ : int 33
## .. ..$ : int 34
## .. ..$ : int 35
## .. ..$ : int 36
## .. ..$ : int 37
## .. ..$ : int 38
## .. ..$ : int 39
## .. ..$ : int 40
## .. ..@ ptype: int(0)
## ..- attr(*, ".drop")= logi TRUE
```

```
# Display values
head(tidyData)
```

```
## # A tibble: 6 x 81
## # Groups:   Subjects, Activity [6]
##   Subjects Activity      'mean_tBodyAcc-m~ 'mean_tBodyAcc-m~ 'mean_tBodyAcc--~
##     <int> <chr>          <dbl>          <dbl>          <dbl>
## 1       1 WALKING          0.266          -0.0183         -0.108
## 2       2 WALKING          0.273          -0.0191         -0.116
## 3       3 WALKING          0.273          -0.0179         -0.106
## 4       4 WALKING          0.277          -0.0133         -0.106
## 5       4 WALKING_UPSTAIRS 0.270          -0.0171         -0.110
## 6       5 WALKING          0.279          -0.0155         -0.106
## # ... with 76 more variables: mean_tBodyAcc-std()-X <dbl>,
## #   mean_tBodyAcc-std()-Y <dbl>, mean_tBodyAcc-std()-Z <dbl>,
## #   mean_tGravityAcc-mean()-X <dbl>, mean_tGravityAcc-mean()-Y <dbl>,
## #   mean_tGravityAcc-mean()-Z <dbl>, mean_tGravityAcc-std()-X <dbl>,
## #   mean_tGravityAcc-std()-Y <dbl>, mean_tGravityAcc-std()-Z <dbl>,
## #   mean_tBodyAccJerk-mean()-X <dbl>, mean_tBodyAccJerk-mean()-Y <dbl>,
## #   mean_tBodyAccJerk-mean()-Z <dbl>, mean_tBodyAccJerk-std()-X <dbl>, ...
```