

```

# https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip
#
# You should create one R script called run_analysis.R that does the following.
#
# 1. Merges the training and the test sets to create one data set.
#
# 2. Extracts only the measurements on the mean and standard deviation for each measurement.
#
# 3. Uses descriptive activity names to name the activities in the data set
#
# 4. Appropriately labels the data set with descriptive variable names.
#
# 5. From the data set in step 4, creates a second, independent tidy data set with the average of each
library(downloader)
library(readr)
library(tidyverse)

```

```

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v dplyr 1.0.9
## v tibble 3.1.7       v stringr 1.4.0
## v tidyr 1.2.0        v forcats 0.5.1
## v purrr 0.3.4
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()      masks stats::lag()

```

```

options(scipen = 999)

# helper function to bind columns of different lengths and fill shorter column with filler NA's
bind_cols_fill <- function(x, y){
  if(nrow(x) != nrow(y))
  {
    if(nrow(x) < nrow(y))
    {
      while(nrow(x) != nrow(y))
      {
        x[nrow(x)+1,] <- NA
      }
    }
    if(nrow(y) < nrow(x))
    {
      while(nrow(x) != nrow(y))
      {
        y[nrow(y)+1,] <- NA
      }
    }
  }
  cbind(x, y)
}

# Changes strings in scientific notation into dbls
from_scientific_notation <- function(x)
{

```

```

if(is.na(x))
{
  return(NA)
}
parts <- str_split(x, "e", simplify = TRUE)
c <- parts[1]
c <- gsub("[^0-9.-]", "", c)
e <- parts[2]
c <- as.numeric(c)
e <- as.numeric(e)
#y <- b*10^p
y <- c*10^e
y
}

# Creates List of strings in scientific notation
create_as_list <- function(x)
{
  my_list <- strsplit(x, " +")
}

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

# downloads and unzips dataset from given url
download(url, dest="dataset.zip", mode="wb")
unzip ("dataset.zip", exdir = ".")

# import datasets
y_test <- read_csv("UCI HAR Dataset/test/y_test.txt", col_names = "y_test")

## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## dbl (1): y_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

glimpse(y_test)

## Rows: 2,947
## Columns: 1
## $ y_test <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~

x_test <- read_csv("UCI HAR Dataset/test/x_test.txt", col_names = "x_test")

## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): x_test
##

```

```
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
glimpse(x_test)
```

```
## Rows: 2,947
## Columns: 1
## $ x_test <chr> "2.5717778e-001 -2.3285230e-002 -1.4653762e-002 -9.3840400e-001~
```

```
x_test %>% count %>% print
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  2947
```

```
glimpse(x_test)
```

```
## Rows: 2,947
## Columns: 1
## $ x_test <chr> "2.5717778e-001 -2.3285230e-002 -1.4653762e-002 -9.3840400e-001~
```

```
#Looks like x_test is a list created by a string of scientific notation values.
```

```
subject_test <- read_csv("UCI HAR Dataset/test/subject_test.txt", col_names = "subject_test")
```

```
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## dbl (1): subject_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
#glimpse(subject_test)
```

```
subject_test %>% count %>% print
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  2947
```

```
x_y_test <- bind_cols_fill(x_test, y_test)
test_df <- bind_cols_fill(x_y_test, subject_test)
#glimpse(test_df)
#glimpse(test_df)
```

```
y_train <- read_csv("UCI HAR Dataset/train/y_train.txt", col_names = "y_train")
```

```
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## dbl (1): y_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
#glimpse(y_train)
```

```
x_train <- read_csv("UCI HAR Dataset/train/x_train.txt", col_names = "x_train")
```

```
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): x_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
#glimpse(x_train)
```

```
subject_train <- read_csv("UCI HAR Dataset/train/subject_train.txt", col_names = "subject_train")
```

```
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## dbl (1): subject_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
#glimpse(subject_train)
```

```
x_y_train <- bind_cols_fill(x_train, y_train)
train_df <- bind_cols_fill(x_y_train, subject_train)
```

```
full_df <- bind_cols_fill(train_df, test_df)
glimpse(full_df)
```

```
## Rows: 7,352
## Columns: 6
## $ x_train      <chr> "2.8858451e-001 -2.0294171e-002 -1.3290514e-001 -9.95278~
## $ y_train      <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~
## $ subject_train <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ x_test       <chr> "2.5717778e-001 -2.3285230e-002 -1.4653762e-002 -9.38404~
## $ y_test       <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~
## $ subject_test  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
```

```
#get all files from train Inertial Signals
```

```
setwd("~/Getting and Cleaning Data Wk 4/UCI HAR Dataset/train/Inertial Signals")
```

```

file_list <- list.files(path=".", pattern=NULL, all.files=FALSE,
                        full.names=FALSE)
#files from file_list ending in txt
file_list_txt <- list()
for(i in 1:length(file_list))
{
  if(str_detect(file_list[i], ".txt"))
  {
    file_list_txt <- c(file_list_txt, file_list[i])
  }
}
file_list_txt

```

```

## [[1]]
## [1] "body_acc_x_train.txt"
##
## [[2]]
## [1] "body_acc_y_train.txt"
##
## [[3]]
## [1] "body_acc_z_train.txt"
##
## [[4]]
## [1] "body_gyro_x_train.txt"
##
## [[5]]
## [1] "body_gyro_y_train.txt"
##
## [[6]]
## [1] "body_gyro_z_train.txt"
##
## [[7]]
## [1] "total_acc_x_train.txt"
##
## [[8]]
## [1] "total_acc_y_train.txt"
##
## [[9]]
## [1] "total_acc_z_train.txt"

```

```

for (i in 1:length(file_list_txt))
{
  name_string <- as.character(file_list_txt[i])
  name <- sub("\\\\.\\.*", "", name_string)
  temp <- read_csv(file_list_txt[i], col_names = name)
  full_df <- bind_cols_fill(temp, full_df)
}

```

```

## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_acc_x_train
##

```

```

## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_acc_y_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_acc_z_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_gyro_x_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_gyro_y_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_gyro_z_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): total_acc_x_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): total_acc_y_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 7352 Columns: 1
## -- Column specification -----
## Delimiter: ","

```

```
## chr (1): total_acc_z_train
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
#glimpse(full_df)

#get all files from test Inertial Signals
setwd("~/Getting and Cleaning Data Wk 4/UCI HAR Dataset/test/Inertial Signals")
file_list <- list.files(path=".", pattern=NULL, all.files=FALSE,
                        full.names=FALSE)
#files from file_list ending in txt
file_list_txt <- list()
for(i in 1:length(file_list))
{
  if(str_detect(file_list[i], ".txt"))
  {
    file_list_txt <- c(file_list_txt, file_list[i])
  }
}
file_list_txt
```

```
## [[1]]
## [1] "body_acc_x_test.txt"
##
## [[2]]
## [1] "body_acc_y_test.txt"
##
## [[3]]
## [1] "body_acc_z_test.txt"
##
## [[4]]
## [1] "body_gyro_x_test.txt"
##
## [[5]]
## [1] "body_gyro_y_test.txt"
##
## [[6]]
## [1] "body_gyro_z_test.txt"
##
## [[7]]
## [1] "total_acc_x_test.txt"
##
## [[8]]
## [1] "total_acc_y_test.txt"
##
## [[9]]
## [1] "total_acc_z_test.txt"
```

```
for (i in 1:length(file_list_txt))
{
  name_string <- as.character(file_list_txt[i])
  name <- sub("\\\\.\\.*", "", name_string)
```

```

temp <- read_csv(file_list_txt[i], col_names = name)
full_df <- bind_cols_fill(temp, full_df)
}

```

```

## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_acc_x_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_acc_y_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_acc_z_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_gyro_x_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_gyro_y_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): body_gyro_z_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): total_acc_x_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

```



```
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): total_acc_y_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## Rows: 2947 Columns: 1
## -- Column specification -----
## Delimiter: ","
## chr (1): total_acc_z_test
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
glimpse(full_df)
```

```
## Rows: 7,352
## Columns: 24
## $ total_acc_z_test <chr> "2.3779770e-002 7.6292710e-002 1.4747540e-001 1.3~
## $ total_acc_y_test <chr> "-2.6979590e-001 -2.8002500e-001 -2.9266310e-001 -2.~
## $ total_acc_x_test <chr> "1.0412160e+000 1.0418030e+000 1.0390860e+000 1.0~
## $ body_gyro_z_test <chr> "1.3652790e-001 9.7622390e-002 8.3555780e-002 2.2~
## $ body_gyro_y_test <chr> "5.3134920e-001 7.2106850e-001 5.2032840e-001 3.7~
## $ body_gyro_x_test <chr> "4.3746370e-001 4.6826410e-001 4.9825740e-001 4.7~
## $ body_acc_z_test <chr> "1.0682620e-001 1.5245490e-001 2.1684620e-001 2.0~
## $ body_acc_y_test <chr> "-2.9399040e-002 -3.9728670e-002 -5.2405860e-002 -5.~
## $ body_acc_x_test <chr> "1.1653150e-002 1.3109090e-002 1.1268850e-002 2.7~
## $ total_acc_z_train <chr> "1.0293410e-001 1.0568720e-001 1.0210250e-001 1.0~
## $ total_acc_y_train <chr> "-1.2321670e-001 -1.2687560e-001 -1.2400370e-001 -1.~
## $ total_acc_x_train <chr> "1.0128170e+000 1.0228330e+000 1.0220280e+000 1.0~
## $ body_gyro_z_train <chr> "2.2858640e-002 1.0315720e-002 1.3249690e-002 1.7~
## $ body_gyro_y_train <chr> "6.6013620e-002 4.2698970e-002 7.4850180e-002 5.7~
## $ body_gyro_x_train <chr> "3.0191220e-002 4.3710710e-002 3.5687800e-002 4.0~
## $ body_acc_z_train <chr> "5.5560680e-002 5.5124830e-002 4.8404730e-002 4.9~
## $ body_acc_y_train <chr> "1.0766810e-002 6.5794800e-003 8.9288780e-003 7.4~
## $ body_acc_x_train <chr> "1.8085150e-004 1.0138560e-002 9.2755740e-003 5.0~
## $ x_train <chr> "2.8858451e-001 -2.0294171e-002 -1.3290514e-001 -9.9~
## $ y_train <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5~
## $ subject_train <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ x_test <chr> "2.5717778e-001 -2.3285230e-002 -1.4653762e-002 -9.3~
## $ y_test <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5~
## $ subject_test <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2~
```

```
from_scientific_notation_v <- base::Vectorize(from_scientific_notation)
```

```
full_df_numeric <- full_df %>%
```

```
  mutate(x_test = from_scientific_notation_v(create_as_list(toString(full_df$x_test))),
         x_train = from_scientific_notation_v(create_as_list(toString(full_df$x_train))),
         body_acc_x_train = from_scientific_notation_v(create_as_list(toString(full_df$body_acc_x_train))),
         body_acc_y_train = from_scientific_notation_v(create_as_list(toString(full_df$body_acc_y_train))),
         body_acc_z_train = from_scientific_notation_v(create_as_list(toString(full_df$body_acc_z_train))),
         body_gyro_x_train = from_scientific_notation_v(create_as_list(toString(full_df$body_gyro_x_train))),
```

```

body_gyro_y_train = from_scientific_notation_v(create_as_list(toString(full_df$body_gyro_y_train))
body_gyro_z_train = from_scientific_notation_v(create_as_list(toString(full_df$body_gyro_z_train))
total_acc_x_train = from_scientific_notation_v(create_as_list(toString(full_df$total_acc_x_train))
total_acc_y_train = from_scientific_notation_v(create_as_list(toString(full_df$total_acc_y_train))
total_acc_z_train = from_scientific_notation_v(create_as_list(toString(full_df$total_acc_z_train))
body_gyro_x_test = from_scientific_notation_v(create_as_list(toString(full_df$body_gyro_x_test))
body_gyro_y_test = from_scientific_notation_v(create_as_list(toString(full_df$body_gyro_y_test))
body_gyro_z_test = from_scientific_notation_v(create_as_list(toString(full_df$body_gyro_z_test))
total_acc_x_test = from_scientific_notation_v(create_as_list(toString(full_df$total_acc_x_test))
total_acc_y_test = from_scientific_notation_v(create_as_list(toString(full_df$total_acc_y_test))
total_acc_z_test = from_scientific_notation_v(create_as_list(toString(full_df$total_acc_z_test))
body_acc_x_test = from_scientific_notation_v(create_as_list(toString(full_df$body_acc_x_test))
body_acc_y_test = from_scientific_notation_v(create_as_list(toString(full_df$body_acc_y_test))
body_acc_z_test = from_scientific_notation_v(create_as_list(toString(full_df$body_acc_z_test))
)

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

```

```

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first

```

```
## element will be used

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first
## element will be used
```

```
glimpse(full_df_numeric)
```

```
## Rows: 7,352
## Columns: 24
## $ total_acc_z_test <dbl> 101269496, 101269496, 101269496, 101269496, 10126949~
## $ total_acc_y_test <dbl> -0.004273516, -0.004273516, -0.004273516, -0.0042735~
## $ total_acc_x_test <dbl> 11.4642, 11.4642, 11.4642, 11.4642, 11.4642, 11.4642~
## $ body_gyro_z_test <dbl> 7896968582, 7896968582, 7896968582, 7896968582, 7896~
## $ body_gyro_y_test <dbl> 86310778, 86310778, 86310778, 86310778, 86310778, 86~
## $ body_gyro_x_test <dbl> 210660.5, 210660.5, 210660.5, 210660.5, 210660.5, 21~
## $ body_acc_z_test <dbl> 35.74594, 35.74594, 35.74594, 35.74594, 35.74594, 35~
## $ body_acc_y_test <dbl> -0.0003129437, -0.0003129437, -0.0003129437, -0.0003~
## $ body_acc_x_test <dbl> 23.84253, 23.84253, 23.84253, 23.84253, 23.84253, 23~
## $ total_acc_z_train <dbl> 11.7336, 11.7336, 11.7336, 11.7336, 11.7336, 11.7336~
## $ total_acc_y_train <dbl> -0.0663611, -0.0663611, -0.0663611, -0.0663611, -0.0~
## $ total_acc_x_train <dbl> 10.6749, 10.6749, 10.6749, 10.6749, 10.6749, 10.6749~
## $ body_gyro_z_train <dbl> 24.58229, 24.58229, 24.58229, 24.58229, 24.58229, 24~
## $ body_gyro_y_train <dbl> 122894, 122894, 122894, 122894, 122894, 122894, 1228~
## $ body_gyro_x_train <dbl> 70949.88, 70949.88, 70949.88, 70949.88, 70949.88, 70~
## $ body_acc_z_train <dbl> 1808217, 1808217, 1808217, 1808217, 1808217, 1808217~
## $ body_acc_y_train <dbl> 4088529, 4088529, 4088529, 4088529, 4088529, 4088529~
## $ body_acc_x_train <dbl> 18.67145, 18.67145, 18.67145, 18.67145, 18.67145, 18~
## $ x_train <dbl> 0.02696845, 0.02696845, 0.02696845, 0.02696845, 0.02~
## $ y_train <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5~
## $ subject_train <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ x_test <dbl> 0.01207009, 0.01207009, 0.01207009, 0.01207009, 0.01~
## $ y_test <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5~
## $ subject_test <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2~
```

```
means_and_sd <- full_df_numeric %>%
  summarise_all(list(mean = mean,
                    sd = sd))

glimpse(means_and_sd)
```

```
## Rows: 1
## Columns: 48
## $ total_acc_z_test_mean <dbl> 101269496
## $ total_acc_y_test_mean <dbl> -0.004273516
## $ total_acc_x_test_mean <dbl> 11.4642
## $ body_gyro_z_test_mean <dbl> 7896968582
## $ body_gyro_y_test_mean <dbl> 86310778
## $ body_gyro_x_test_mean <dbl> 210660.5
## $ body_acc_z_test_mean <dbl> 35.74594
## $ body_acc_y_test_mean <dbl> -0.0003129437
## $ body_acc_x_test_mean <dbl> 23.84253
## $ total_acc_z_train_mean <dbl> 11.7336
## $ total_acc_y_train_mean <dbl> -0.0663611
## $ total_acc_x_train_mean <dbl> 10.6749
## $ body_gyro_z_train_mean <dbl> 24.58229
## $ body_gyro_y_train_mean <dbl> 122894
## $ body_gyro_x_train_mean <dbl> 70949.88
## $ body_acc_z_train_mean <dbl> 1808217
## $ body_acc_y_train_mean <dbl> 4088529
## $ body_acc_x_train_mean <dbl> 18.67145
## $ x_train_mean <dbl> 0.02696845
## $ y_train_mean <dbl> 3.643362
## $ subject_train_mean <dbl> 17.41308
## $ x_test_mean <dbl> 0.01207009
## $ y_test_mean <dbl> NA
## $ subject_test_mean <dbl> NA
## $ total_acc_z_test_sd <dbl> 0
## $ total_acc_y_test_sd <dbl> 0
## $ total_acc_x_test_sd <dbl> 0
## $ body_gyro_z_test_sd <dbl> 0
## $ body_gyro_y_test_sd <dbl> 0
## $ body_gyro_x_test_sd <dbl> 0
## $ body_acc_z_test_sd <dbl> 0
## $ body_acc_y_test_sd <dbl> 0
## $ body_acc_x_test_sd <dbl> 0
## $ total_acc_z_train_sd <dbl> 0
## $ total_acc_y_train_sd <dbl> 0
## $ total_acc_x_train_sd <dbl> 0
## $ body_gyro_z_train_sd <dbl> 0
## $ body_gyro_y_train_sd <dbl> 0
## $ body_gyro_x_train_sd <dbl> 0
## $ body_acc_z_train_sd <dbl> 0
## $ body_acc_y_train_sd <dbl> 0
## $ body_acc_x_train_sd <dbl> 0
## $ x_train_sd <dbl> 0
## $ y_train_sd <dbl> 1.744802
## $ subject_train_sd <dbl> 8.975143
```

```
## $ x_test_sd          <dbl> 0
## $ y_test_sd          <dbl> NA
## $ subject_test_sd    <dbl> NA
```

```
tidy_means <- full_df_numeric %>%
  summarise_all(list(mean = mean)) %>%
  pivot_longer(cols = everything()) %>%
  rename("variables" = "name") %>%
  rename("mean" = "value")

glimpse(tidy_means)
```

```
## Rows: 24
## Columns: 2
## $ variables <chr> "total_acc_z_test_mean", "total_acc_y_test_mean", "total_acc~
## $ mean      <dbl> 101269496.2773715258, -0.0042735161, 11.4642020893, 78969685~
```