# Fit Tracker Google Analytics Capstone Project

Michael

2022-06-11

## Contents

# Introduction

This is my capstone project for the Google Data Analytics Certificate Program. In this capstone, I am tasked with analyzing a set of data collected from Fitbit users to gain insight into health-tracking devices and using this insight to inform key business questions at the fictional company 'Bellabeat', a health-focused smart device company. Bellbeat is interested in developing their own line of health-focused wearables, and are looking for data driven guidance for their marketing strategy of their new wearable devices.

# Deliverables

There are 6 Deliverables for this Project as Stated in the Task Description:

1) A summary of the business task
2) A description of all data sources used
3) Documentation of the data cleaning process
4) A summary of the analysis
5) Visualizations and key findings
6) Recommendations based on our findings

These deliverables loosely map onto the 6 Phases of Data Analysis: 1) Ask, 2) Prepare, 3) Process, 4) Analyze, 5) Share, and 6) Act.

This report will cover each of these 6 steps of the data analysis process, and provide the corresponding deliverable for each step (eg. 'Phase 1 - Ask' will provide the 'summary of the business task' deliverable)

# PHASE 1 - ASK

Our goal in analyzing this data is to gain insight into how customers use smart devices in order to inform our company's marketing strategy.

To help guide our analysis, we begin by breaking this goal down further into the following questions we would like to answer:

1.1) What are trends in usage?
1.2) What are potential use-cases for similar products?
1.3) How could these trends and use-cases apply to our customers?
1.4) How could these insights influence our marketing strategy?

We revisit these questions in the Phase 6 (Act) section of this document.

# PHASE 2 - PREPARE

In this phase we collect our data sets, setup secure data storage and begin assessing the quality of the data by considering its source, sample-size, bias, reliability, and other factors. We include this summary below:

## 2.1) Where is the Data Stored?

Data is stored in folder system on my PC. I use an R package, sqldf, to query the data in the form of a relational database using SQL syntax.

## 2.2) How is the Data Organized?

The data is separated into data sets for different metrics (eg. calories, steps, intensities, heart rate, etc.) It is further separated within data frames by time (day, hour, minute).

## 2.3) Is the Data Biased or Credible?

This data comes from an external source, so it may not be trustworthy. Some sections might be incredible or biased, eg. weight_log_info.csv contains manual report data. The sample size of 30 is likely too small, considering the population is all Fitbit users. Further demographic information of the users and the rest of the main Fitbut user-base could be useful to determine the extent to which the sample is representative and/or biased.

## 2.4) Discuss the Licensing, Privacy, Security, and Accessibility of the Data.

The data is secure by being password protected, although the data is also open-source. Integrity of data (accuracy, completeness, consistency) - The data is not first party. It comes from an external source so it's quality is suspect. How does it help answer the question? - The data shows users' activity levels, sleep, and calories burned over time. It may be able to identify trends and potential uses for health tracking devices.

## 2.5) What are the Problems with Data?

Our company's target audience is Women, and this data set doesn't contain demographic information, so we cannot assess whether this Fitbit data varies with gender. We will need to keep this in mind during the interpretation of our results. **ROCCC** method for assessing data quality:

**2.5.1) R (Reliable)** This data is from an external source so its reliability is in question. The sample size is also small and there is no demographic information included on the subjects or the Fitbit user population in general.

**2.5.2) O (Original)** This is an original data set collected through users who consented to have their data sent collected.

**2.5.3) C (Comprehensive)** For the users that it does contain data on, this data set has comprehensive data on activity levels etc. Although there is still a lot of information that could be useful for answering our question (eg. What activities do users use their Fitbit during? What features do users most enjoy/use about their Fitbit).

**2.5.4) C (Current)** This data is from March to May of 2016 and is thus not current since Fitbit technology and products have advanced considerably since then. For example, the Fitbit Versa 3 was released in August 2020 and new models are generally released every year or two.

**2.5.5) C (Cited)** Citation information appears to be unavailable for this data.

**Note -** After some exploratory analysis of the given data set, we find an interesting trend between activity intensity and caloric expenditure which we want to explore further. At this point, we returned to the Prepare Phase to find additional data to help us gain insight into this trend. We preform a similar analysis of this supplemental data set that is not included for brevity.

# PHASE 3 - PROCESS

In this phase we clean and format the data.

## 3.1) Import Packages

```
## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.4.1     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 1.0.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## 3.2) Load Data

The 18 data tables are loaded efficiently into a list using the 'lapply()' function.

```
setwd('C:\\Users\\micha\\Desktop\\Coding Projects (older)\\RStudio Scripts and Data\\Google Capstone Ana
files = list.files(pattern = "merged.csv$") #Generates a list of file names
data <- lapply(files, read.csv) #Loads all the files into a list
names(data) <- gsub("\\.csv$", "", files) #Renames data frames with their original names
```

The following 3 steps are performed together in a for-loop. 3.3) Renaming Columns, 3.4) Checking for Missing Data and 3.5) Checking for Duplicates

## 3.3) Renaming Columns

The date/time columns do not have consistent naming across data tables, so we changed the name of all of them to 'TIME' so I can easily access them later via SQL queries.

### 3.4) Checking for Missing Data

For each data frame, I check each column for missing data and identify the columns with missing data. Only the 'Fat' column from the 'weightLogInfo_merged' data frame had missing data. In this column most of the data was missing so it was excluded from our analysis.

### 3.5) Checking for Duplicates

I check for duplicates using the 'sqldf' package to write a SQL query to create a new column that combines participant ID with the date and time of observation to create a unique identifier for each data point. I then use another SQL query to check if the number of distinct observations matches the number of total observations. There were no instances of repeated data.

```r
for (i in 1:length(data)) {
  # check for repeat data
  colnames(data[[i]])[2] <- 'TIME' # rename all the time data columns to "TIME" so I can SQL query them
  df <- data[[i]] # create temporary df to use in query
  time_Id_combo <- sqldf("SELECT Id||' - '||TIME FROM df") # combine ID and TIME into a separate variab
  uniq <- sqldf("SELECT DISTINCT * FROM time_Id_combo") # select distinct values from the id - TIME com
  if (length((time_Id_combo[[1]])) != length(uniq[[1]])) { # compare for duplicates
    cat('There are', length(time_Id_combo[[1]]) - length(uniq[[1]]), 'repeat time-id combos', '\n') # t
  }
  # check for Na's
  for (col in data[[i]]) {
    nas <- sum(is.na(col))
    if (nas >= 1) { # There are several NA's in the "Fat" attribute of the weightLogInfo data frame
      cat('There are', nas, 'NAs in this column', '\n')
    }
  }
  summary(data[[i]]) # check summary of each data frame
}
```

```
## There are 543 repeat time-id combos
## There are 3 repeat time-id combos
## There are 65 NAs in this column
```

# PHASE 4 - ANALYZE

In this phase we identify and investigate trends, patterns, and interesting findings in the data.

During analysis we separated this phase into 2 sub-processes: Exploratory Analysis and In-depth Analysis. However, much of the exploratory analysis that did not lead to major insights is excluded from this report for brevity's sake.

We found 3 primary potential insights: 4.1) Exploration of Heart Rate Data, 4.2) Investigation of Number of Steps on Calories Burned, and 4.2) Investigation of Activity Intensity on Calories Burned.

### 4.1) Potential Insight: Exploration of Heart Rate Data

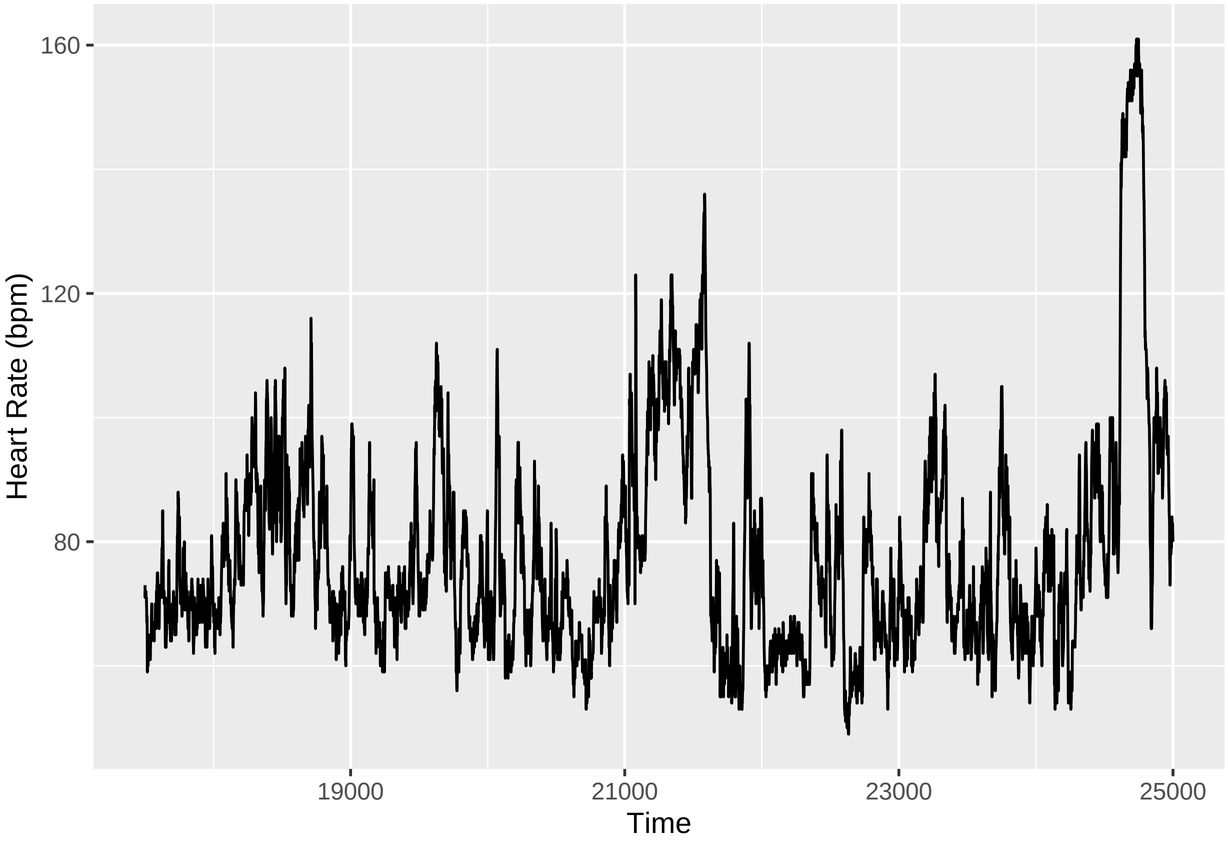
Can time-series heart rate data be used to detect features of interest, such as spikes and/or troughs that could warn the user of potential medical emergencies?

First, use SQL to select a sample user's heart rate data

```
hr_over_time <- sqldf("SELECT *
                      FROM df2
                      WHERE id = 2022484408")
```

There is too much data for one time-series plot, so we will select a snapshot of the data to look at.

```
snapshot <- sqldf("SELECT *
                  FROM hr_over_time
                  WHERE x >= 17500 AND x <= 25000")
ggplot(data = snapshot, aes(x = x, y = Value)) +
  geom_line() +
  xlab('Time') +
  ylab('Heart Rate (bpm)')
```



We can see a wide range of heart rate variability for the participant over time. We see some dramatic peaks, so next, we will investigate whether any abnormally fast or abnormally slow heart rates are measured by the Fitbit over the entire time frame.

```
over_200 <- sqldf("SELECT *
                  FROM hr_over_time
                  WHERE Value >= 200")
under_50 <- sqldf("SELECT *
                  FROM hr_over_time
                  WHERE value <= 45")
head(over_200)
```

```
##           Id                  TIME Value     x
## 1 2022484408 4/21/2016 4:31:20 PM   200 49260
## 2 2022484408 4/21/2016 4:31:30 PM   202 49261
## 3 2022484408 4/21/2016 4:31:40 PM   203 49262
## 4 2022484408 4/21/2016 4:31:50 PM   202 49263
## 5 2022484408 4/21/2016 4:32:00 PM   203 49264
## 6 2022484408 4/21/2016 4:32:10 PM   203 49265
```

```
head(under_50)
```

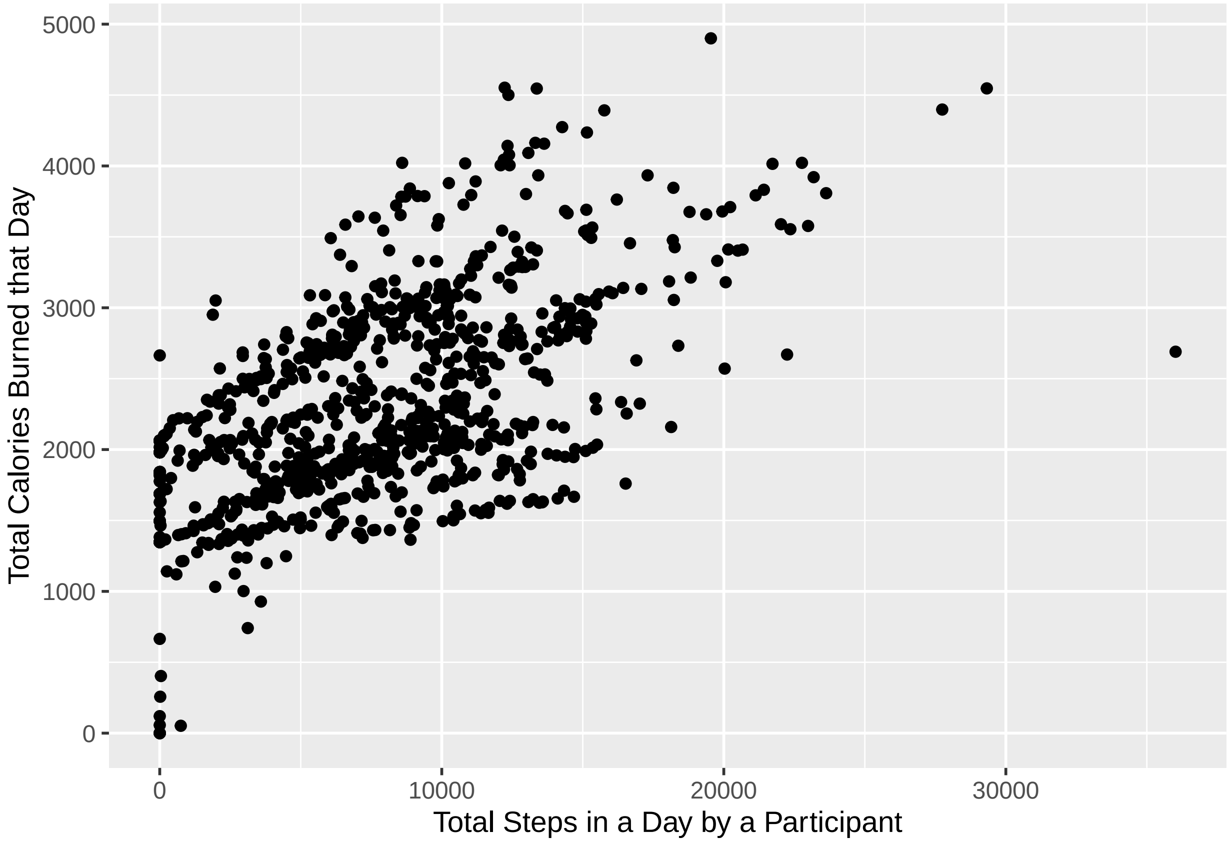
```
##           Id                  TIME Value     x
## 1 2022484408 4/27/2016 1:44:50 PM    45 76315
## 2 2022484408 4/27/2016 1:45:05 PM    45 76316
## 3 2022484408 4/27/2016 1:45:10 PM    42 76317
## 4 2022484408 4/27/2016 1:45:20 PM    41 76318
## 5 2022484408 4/27/2016 1:45:35 PM    41 76319
## 6 2022484408 4/27/2016 1:45:40 PM    42 76320
```

There were times where the participant's heart rate was measured as above 200 and also times when it was below 45. Our product could potentially identify abnormal heart rates and alert the user or even notify medical professionals in emergency situations. This could potentially be a use case for our health tracking products.

## 4.2) Potential Insight: The Relationship Between Step Count and Calories Burned

We speculate that there should be a positive relationship between step count and calories burned. To check this speculation, we create a scatter plot comparing total number of steps for each participant for each day to the corresponding number of calories burned that day.

```
ggplot(data = df1, aes(x = TotalSteps, y = Calories)) +
  geom_point() +
  xlab('Total Steps in a Day by a Participant') +
  ylab('Total Calories Burned that Day')
```

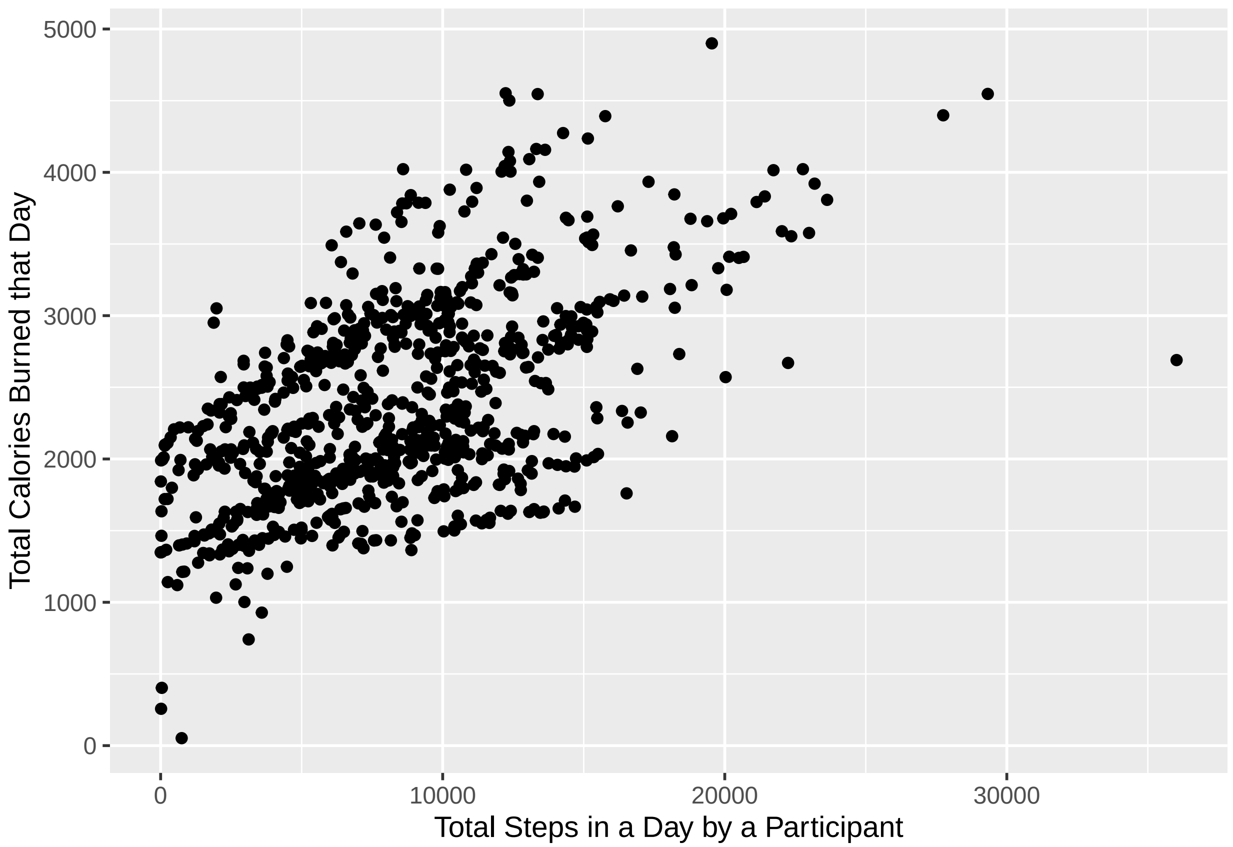


One thing we notice from this graph is that there are data points with 0 steps recorded. After examining this aspect of the data further, we find that many participants have 1 or more days with 0 steps recorded. In fact, a few participants have many days with 0 calories burned. We suspect the most likely reason for this was from participants not wearing the device for a day. We conclude that this can bias our data and and skew averages, so we decide to filter out these data points with a SQL query.

```
df1 <- sqldf("SELECT * FROM df1 WHERE TotalSteps > 0")
```

Now, we re-plot the data.

```
ggplot(data = df1, aes(x = TotalSteps, y = Calories)) +
  geom_point() +
  xlab('Total Steps in a Day by a Participant') +
  ylab('Total Calories Burned that Day')
```



We still see some data points with oddly low caloric expenditure for the day, although these are still non-zero values so that cause is more difficult to determine. We cannot say with confidence whether these data points are erroneous, so we decide to leave them until we can gain more information about them.

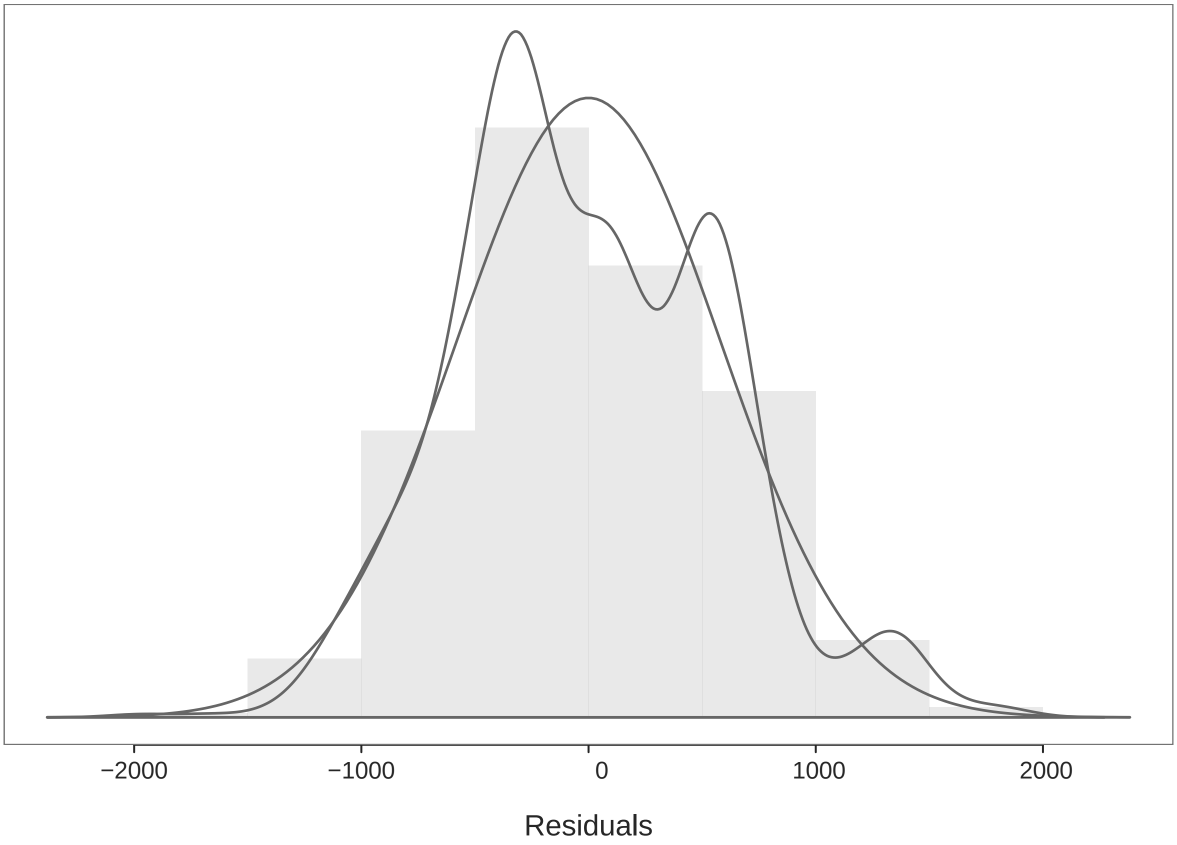
Also from looking at the graph, we see a clear positive relationship between the two, which confirms our speculation. To investigate this further, we create a simple linear regression model of caloric expenditure regressed on number of steps using the 'Regression' function from the 'lessR' package.

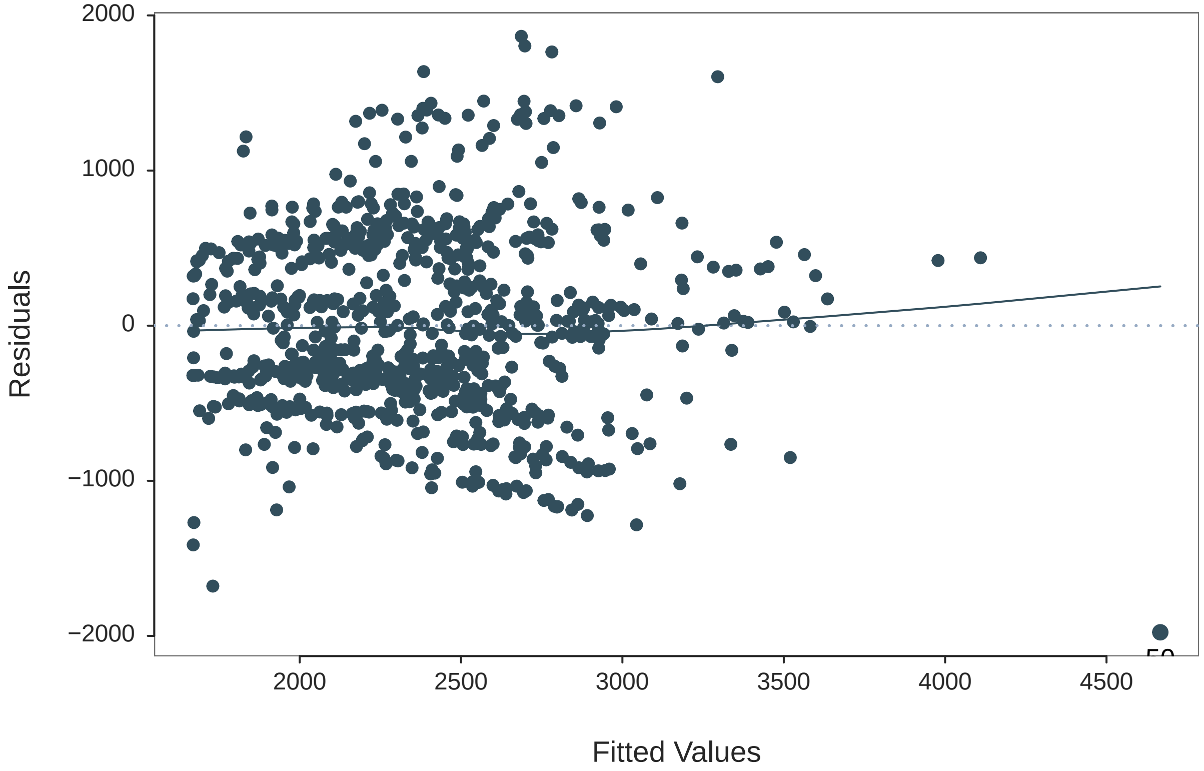
```
library(lessR)
```

```
##
## lessR 4.1.8                         feedback: gerbing@pdx.edu
## --------------------------------------------------------------
## > d <- Read("")   Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Learn about reading, writing, and manipulating data, graphics,
## testing means and proportions, regression, factor analysis,
## customization, and descriptive statistics from pivot tables.
##   Enter:  browseVignettes("lessR")
##
## View changes in this or recent versions of lessR.
##   Enter: help(package=lessR)  Click: Package NEWS
##   Enter: interact()  for access to interactive graphics
##   New function: reshape_long() to move data from wide to long

##
```

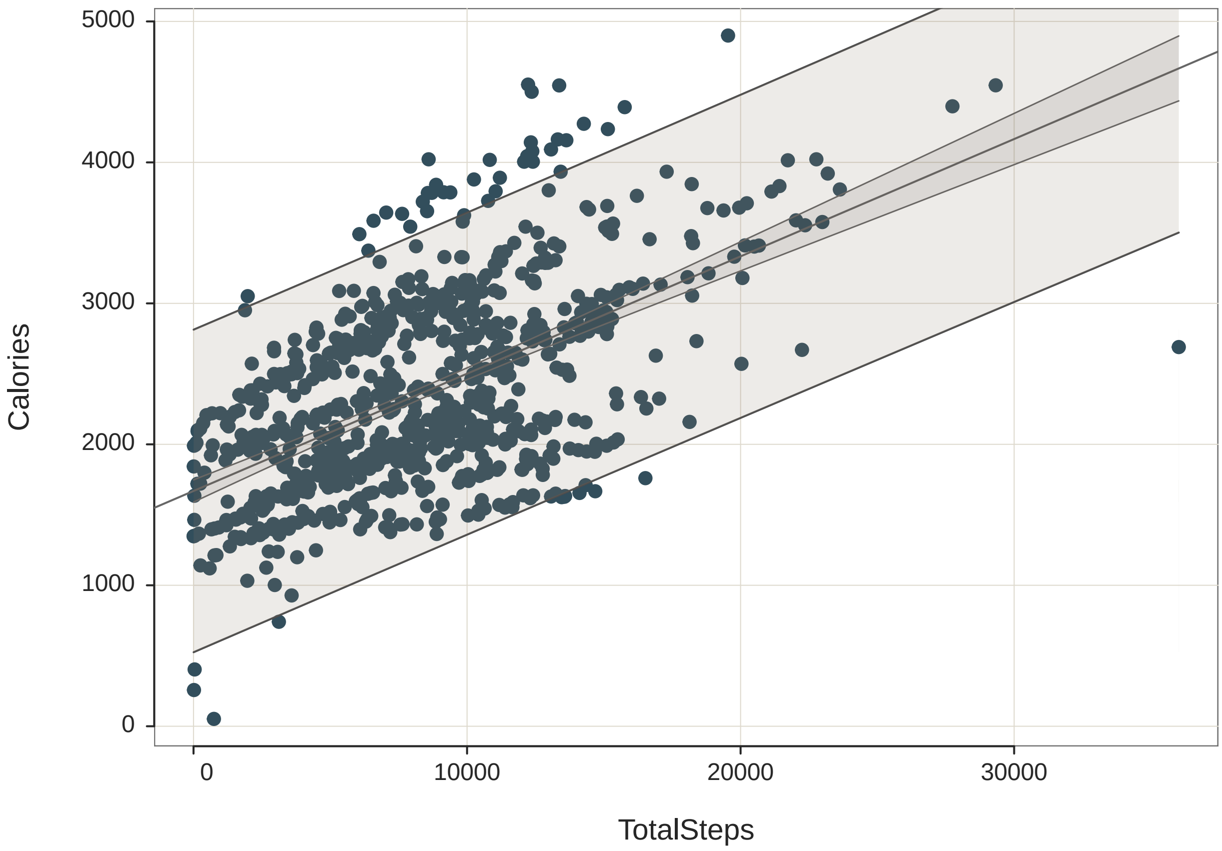
```
## Attaching package: 'lessR'

## The following objects are masked from 'package:dplyr':
##
##     recode, rename
```

```
Regression(Calories ~ TotalSteps, data = df1)
```





Point with largest Cook's Distance of 0.26 is labeled



```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=Calories ~ TotalSteps, data=df1, Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  df1
##
## Response Variable: Calories
## Predictor Variable: TotalSteps
##
## Number of cases (rows) of data:  863
## Number of cases retained for analysis:  863
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)  1668.895     39.977   41.746    0.000    1590.431    1747.359
##  TotalSteps     0.083      0.004   19.936    0.000       0.075       0.091
##
## Standard deviation of Calories: 702.711
##
## Standard deviation of residuals:  581.585 for 861 degrees of freedom
## 95% range of residual variation:  2,282.980 = 2 * (1.963 * 581.585)
```

```
##
## R-squared:  0.316    Adjusted R-squared:  0.315    PRESS R-squared:  0.312
##
## Null hypothesis of all 0 population slope coefficients:
##   F-statistic: 397.447     df: 1 and 861     p-value:  0.000
##
## -- Analysis of Variance
##
##              df        Sum Sq        Mean Sq   F-value   p-value
## Model         1  134432727.904  134432727.904   397.447     0.000
## Residuals   861  291225421.748     338240.908
## Calories    862  425658149.652     493802.958
##
##
##   K-FOLD CROSS-VALIDATION
##
##
##   RELATIONS AMONG THE VARIABLES
##
##              Calories TotalSteps
##     Calories     1.00       0.56
##   TotalSteps     0.56       1.00
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 863 rows of data, or do res_rows="all"]
## ----------------------------------------------------------------
##       TotalSteps Calories   fitted     resid rstdnt dffits cooks
##    50      36019     2690 4666.658 -1976.658 -3.493 -0.719 0.255
##   561      19542     4900 3295.322  1604.678  2.781  0.244 0.030
##   321        746       52 1730.983 -1678.983 -2.905 -0.187 0.017
##   372         17      257 1670.310 -1413.310 -2.443 -0.168 0.014
##   227      22244     2670 3520.203  -850.203 -1.471 -0.156 0.012
##   539      15764     4392 2980.890  1411.110  2.438  0.155 0.012
##   546      13368     4546 2781.477  1764.523  3.052  0.152 0.011
##   312      16520     1760 3043.810 -1283.810 -2.218 -0.151 0.011
##   240         42      403 1672.391 -1269.391 -2.193 -0.150 0.011
##   532      12231     4552 2686.848  1865.152  3.228  0.143 0.010
##   545      12363     4501 2697.834  1803.166  3.119  0.140 0.010
##    16      18134     2159 3178.138 -1019.138 -1.760 -0.138 0.010
##   762      15148     4236 2929.622  1306.378  2.256  0.135 0.009
##   531      14269     4274 2856.465  1417.535  2.448  0.134 0.009
##   319      14687     1667 2891.254 -1224.254 -2.113 -0.121 0.007
##   630      20031     2571 3336.021  -765.021 -1.321 -0.120 0.007
##   325       1982     3051 1833.852  1217.148  2.100  0.120 0.007
##   759      13630     4157 2803.283  1353.717  2.337  0.120 0.007
##   837      29326     4547 4109.618   437.382  0.761  0.119 0.007
##   755      13318     4163 2777.316  1385.684  2.392  0.118 0.007
##
##
##   PREDICTION ERROR
```

```
##
## Data, Predicted, Standard Error of Forecast,
## 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
##    [to see all intervals do pred_rows="all"]
##  ----------------------------------------------
##
##       TotalSteps Calories     pred      sf   pi.lwr   pi.upr    width
##   104          4     1348 1669.228 582.956  525.046 2813.410 2288.363
##   100          8     1349 1669.561 582.955  525.381 2813.741 2288.359
##   616          9     1843 1669.644 582.955  525.465 2813.823 2288.358
## ...
##   582       8301     2783 2359.765 581.922 1217.614 3501.916 2284.302
##   264       8314     1971 2360.847 581.922 1218.696 3502.998 2284.302
##   535       8330     3192 2362.178 581.922 1220.027 3504.329 2284.302
## ...
##   837      29326     4547 4109.618 588.493 2954.570 5264.666 2310.096
##    50      36019     2690 4666.658 593.300 3502.175 5831.142 2328.967
##
## ---------------------------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: Reg Line, Confidence & Prediction Intervals
## ---------------------------------------------------
```

The output of the 'Regression()' function shows that there is a strong correlation of 0.56 between calories burned and total steps. The 3 plots referenced in the output are also plotted, which allows us to check our statistical assumptions (1. There is a linear relationship between the variables, 2. The data forms a bivariate normal distribution, 3. There are few bivariate outliers, 4. The residuals of the model are normally distributed, 5. The average residual error should be near 0 across values of the predictor variable, and 6. The variance of residuals should be even across the range of fitted values).

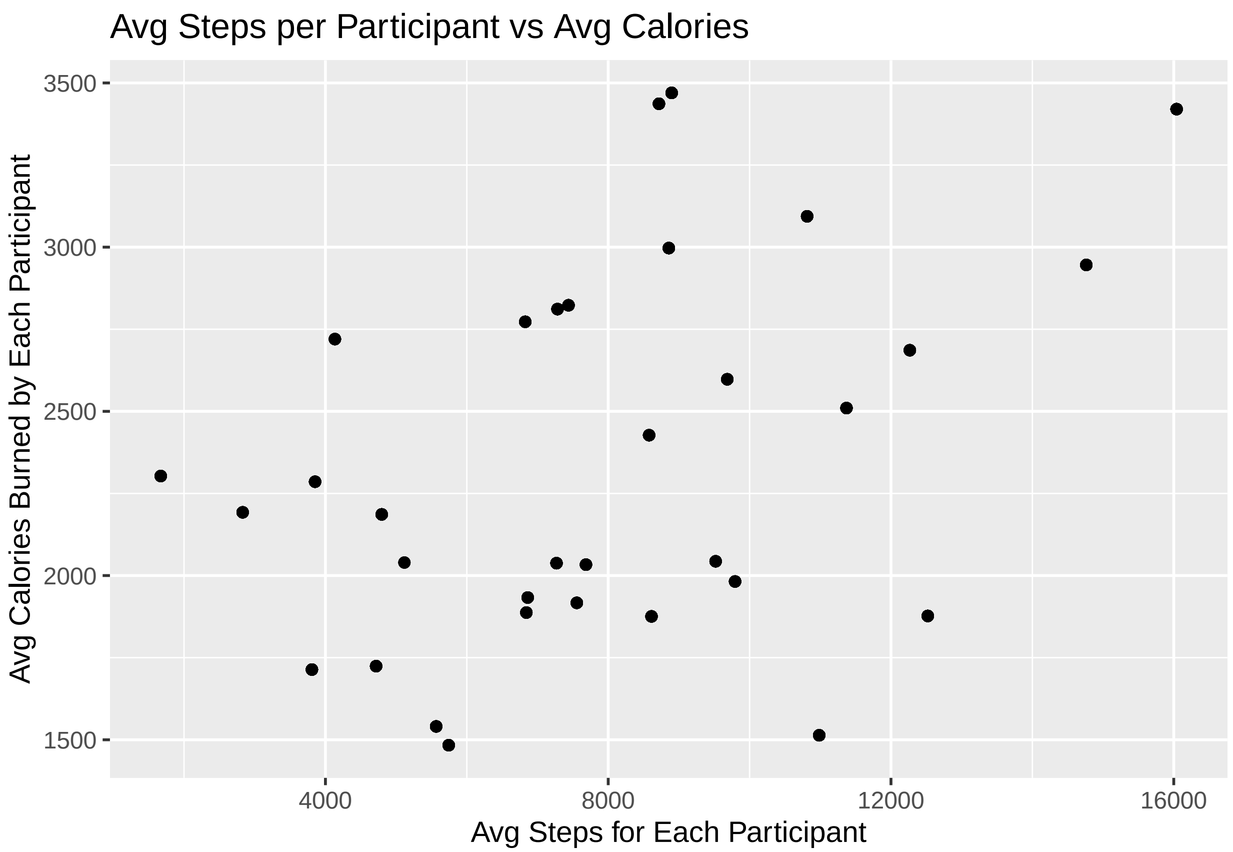
We further investigate this trend by looking at 1) the relationship between the average steps and average calories burned for each participant (averaged across dates) and 2) the relationship between the average steps and average calories burned for each day (average across participants).

To do this, we create 2 new dataframes, one that is grouped by participant id, and one that is grouped by date.

```r
df1_ids <- df1 %>%
  group_by(df1$Id) %>%
  mutate(avgCals = mean(Calories), avgSteps = mean(TotalSteps))

df1_dates <- df1 %>%
  group_by(df1$TIME) %>%
  mutate(avgCals = mean(Calories), avgSteps = mean(TotalSteps))
```

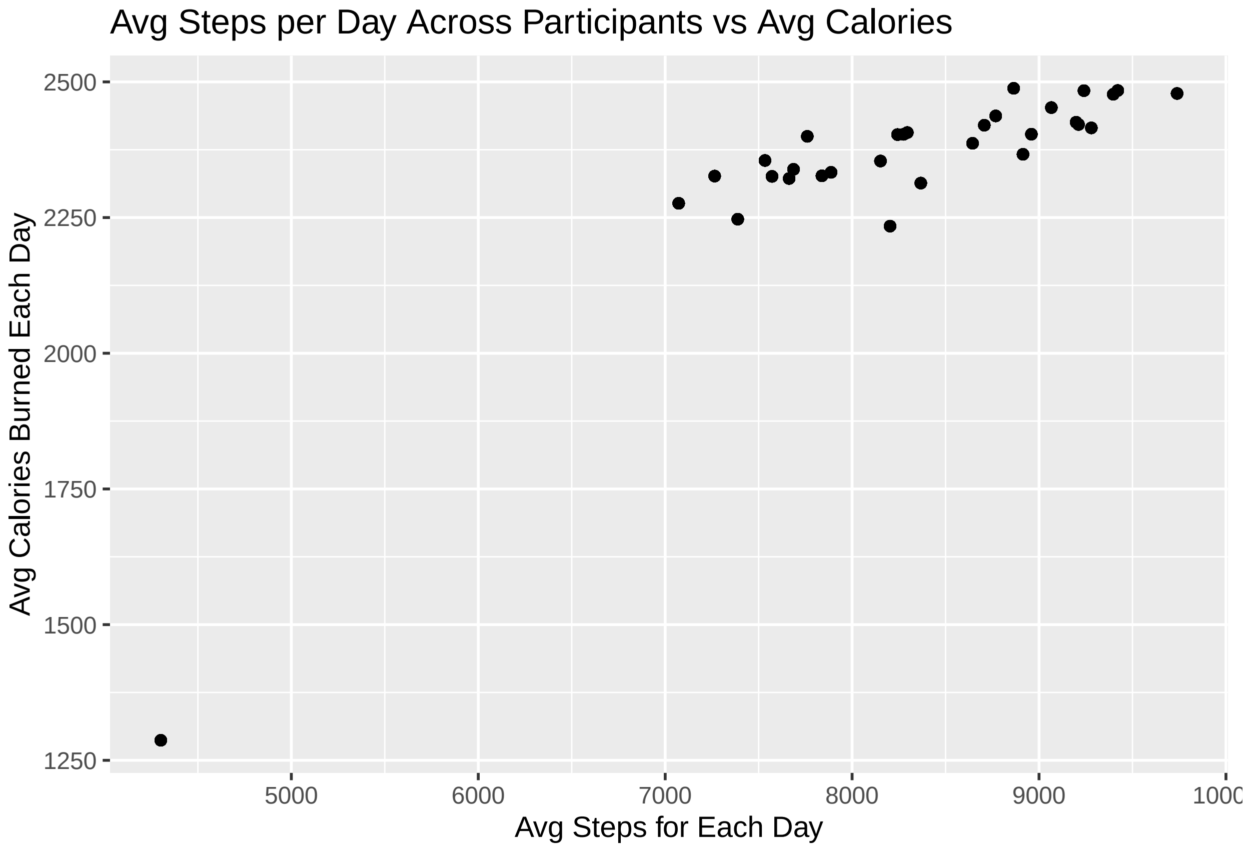
Next, using the dataframe grouped by participant, we create a scatter plot comparing the average calories burned with the average number of total steps per participant

```r
ggplot(data = df1_ids, aes(x = avgSteps, y = avgCals)) +
  geom_point() +
  xlab('Avg Steps for Each Participant') +
  ylab('Avg Calories Burned by Each Participant') +
  ggtitle('Avg Steps per Participant vs Avg Calories')
```



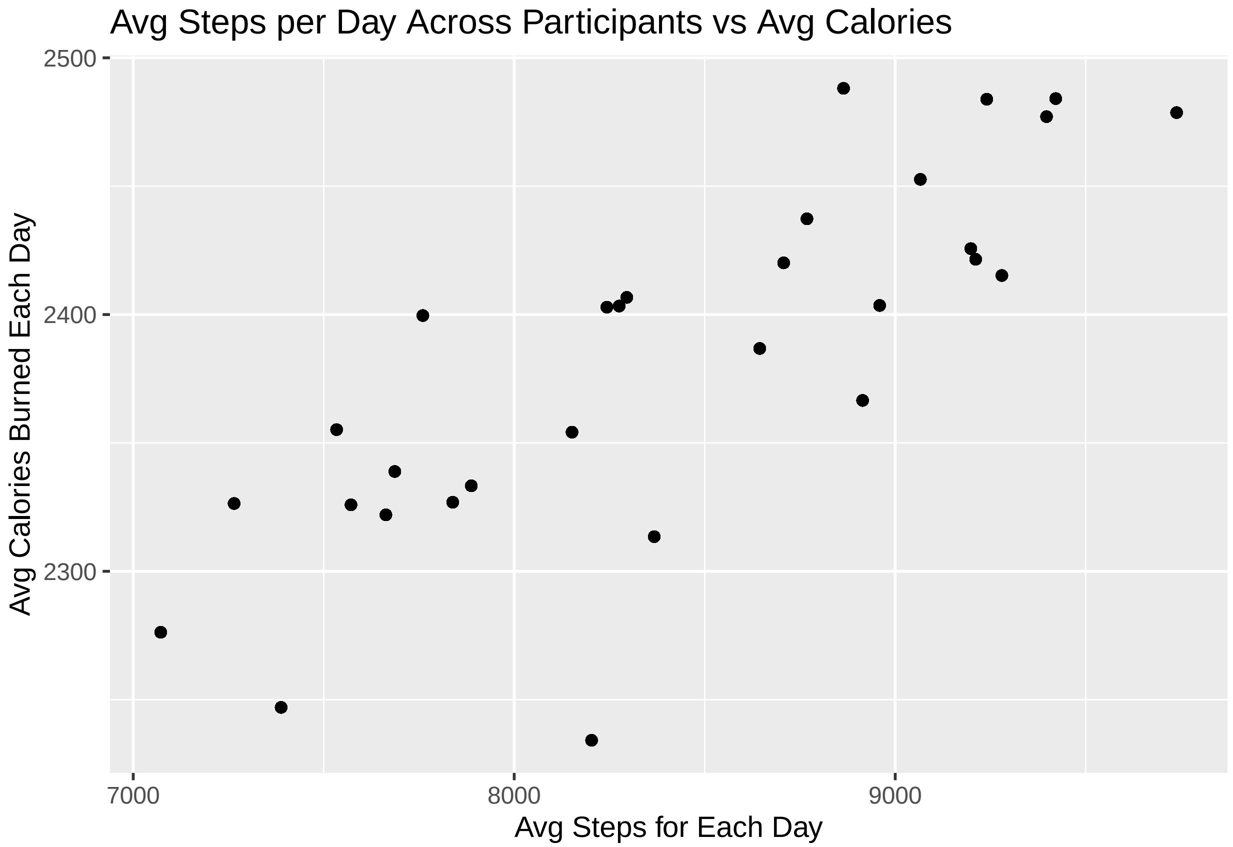
Using the dataframe grouped by date, we create a similar scatter plot:

```
ggplot(data = df1_dates, aes(x = avgSteps, y = avgCals)) +
  geom_point() +
  xlab('Avg Steps for Each Day') +
  ylab('Avg Calories Burned Each Day') +
  ggtitle("Avg Steps per Day Across Participants vs Avg Calories")
```



Noteably, there is an outlier for one of the days with a very low average for both steps and calories burned. This day occured on the last day of the study, and contains many participants who logged an unusually low number of steps and calories burned. We speculated that data collection only occured for part of the last day of the study, leading to vastly lower numbers of both calories burned and total steps. Indeed, upon checking the hourly time series data, we can see that measurements on the last day cease at 2 PM, instead of continuing until midnight. Because this data point likely introduces bias into our data set since it only contains data from the first part of the day, we decide to remove it for our analysis. Additionally, this point greatly enlarges the axes of our graph, so we remove it and re-plot the data below.

```r
df1_dates_no_outlier <- df1_dates %>% filter(TIME != '5/12/2016')
ggplot(data = df1_dates_no_outlier, aes(x = avgSteps, y = avgCals)) +
  geom_point() +
  xlab('Avg Steps for Each Day') +
  ylab('Avg Calories Burned Each Day') +
  ggtitle("Avg Steps per Day Across Participants vs Avg Calories")
```



Finally, we compute the correlations between average number of steps and calories burned for each of the 2 data frames.

```
cor(df1_ids$avgCals, df1_ids$avgSteps)
```

```
## [1] 0.4170512
```

```
cor(df1_dates$avgCals, df1_dates$avgSteps)
```

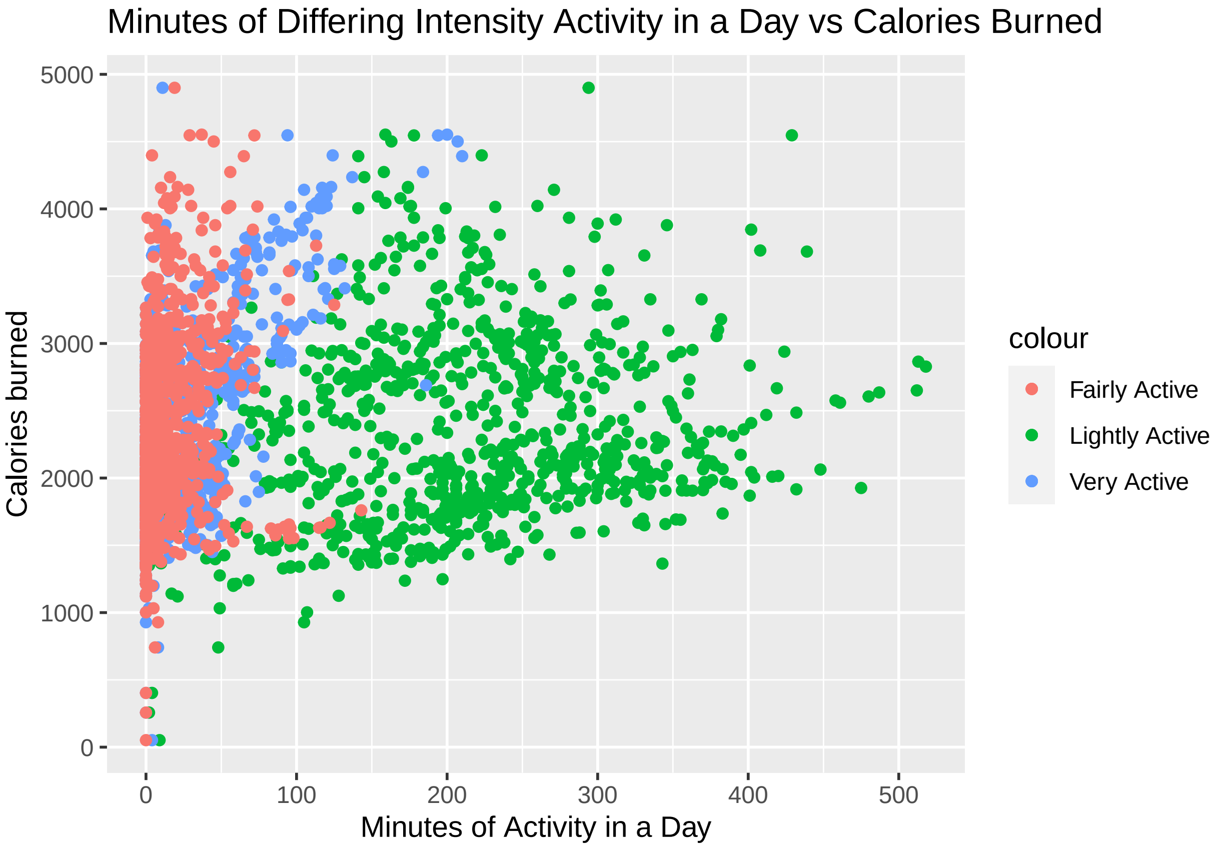
```
## [1] 0.8247289
```

We see strong correlations between caloric expenditure and number of steps both when averaged by day and when averaged by participant (0.42 and 0.82 respectively). Although the correlation is much stronger when averaged across participants for each day. This indicates that there is a strong relationship between total steps in a day and total calories burned for that day. We will use this info later in Phase 6 (Act) section of this report to guide our recommendations.

### 4.3) Potential Insight: Investigating Various Activity Levels on Calories Burned

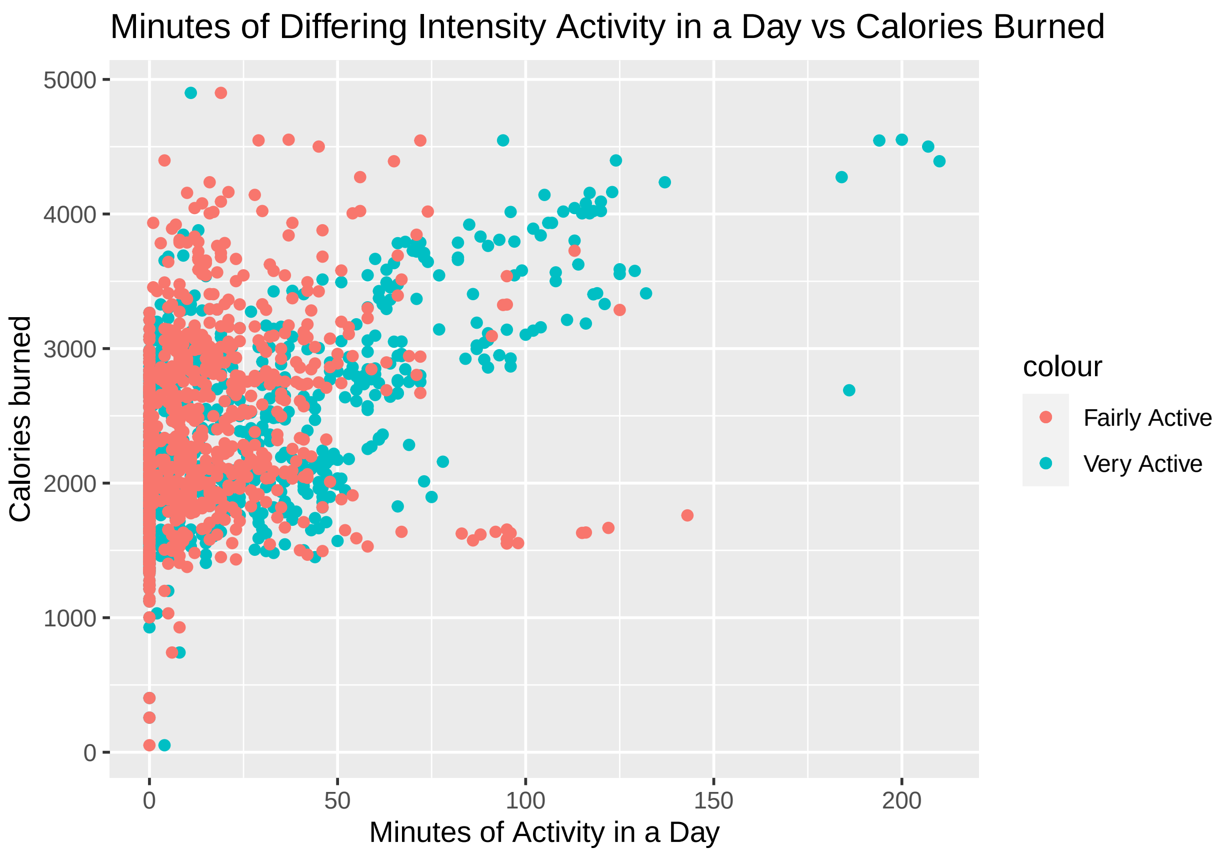
First, we create a scatter plot with all 3 of the activity levels vs. calories burned.

```
ggplot(data = df1, aes(x=LightlyActiveMinutes, y=Calories, color = 'Lightly Active')) +
  labs(x = "Minutes of Activity in a Day", y = "Calories burned") +
  ggtitle("Minutes of Differing Intensity Activity in a Day vs Calories Burned") +
  geom_point() +
  geom_point(data = df1, aes(x=VeryActiveMinutes, y=Calories, color = "Very Active")) +
  geom_point(data = df1, aes(x=FairlyActiveMinutes, y=Calories, color = "Fairly Active"))
```



The graph is a bit crowded, so next we look at just the medium and high intensity data points. We also noted that there are a number of points that show 0 calories burned and/or 0 minutes of activity in a day. We speculate that these may be measurement errors (eg. A user forgot to wear their device for a day).

```
ggplot(data = df1, aes(x=VeryActiveMinutes, y=Calories, color = 'Very Active')) +
  labs(x = "Minutes of Activity in a Day", y = "Calories burned") +
  ggtitle("Minutes of Differing Intensity Activity in a Day vs Calories Burned") +
  geom_point() +
  geom_point(data = df1, aes(x=FairlyActiveMinutes, y=Calories, color = 'Fairly Active'))
```



We notice a positive relationship between minutes of activity and calories burned for each of the 3 intensity levels. Next, we measure the correlations between caloric expenditure and each of the 3 activity levels.

```
cor(df1$LightlyActiveMinutes, df1$Calories) # Cor of .18
```

```
## [1] 0.1817302
```

```
cor(df1$FairlyActiveMinutes, df1$Calories) # Cor of .26
```

```
## [1] 0.264758
```

```
cor(df1$VeryActiveMinutes, df1$Calories) # Cor of .61
```

```
## [1] 0.6124746
```

We find that the correlation between caloric expenditure and highly active minutes is over twice as strong as the other 2 correlations. We identified this as an interesting trend and wondered if high intensity activity is particularly beneficial for burning calories. We decided to investigate it further by finding another data set that would help us answer this question. This data set is discussed below.

row_nums contains all the row numbers of the data set that pertain to the question we are investigating. This selects for only those rows that contain information on how caloric expenditure varies across different intensities of the same activity.

```
row_nums = c(2:7, 15, 14, 18, 17, 21:24, 26,27, 36, 37, 38:48, 96, 95, 116:118, 148, 149, 152:155, 167:
row_nums <- as.data.frame(row_nums) # convert to dataframe to be used in sqldf()
```

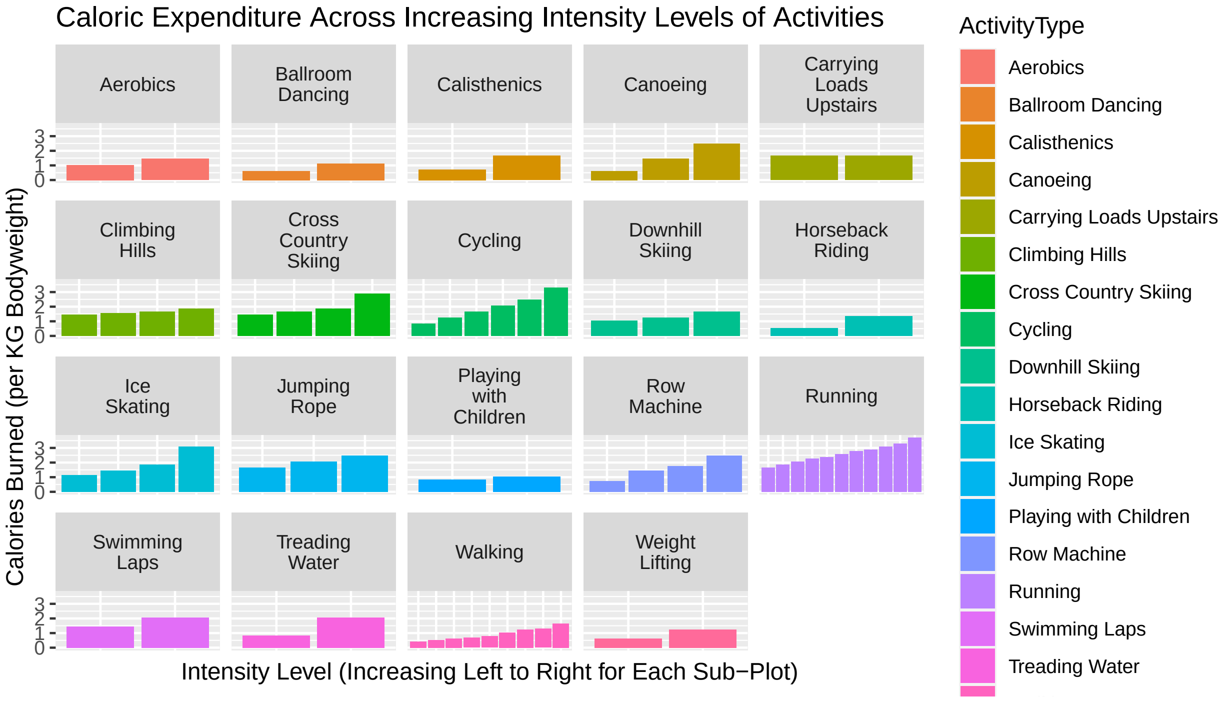
We then write a SQL query that selects only those rows of interest from the original data frame (we select only the row numbers from the data set that are contained in row_nums).

```
# Select the rows that contain activities that are stratified by intensity level

# This 1st query creates a row_num attribute for the dataframe that is used in the 2nd query
intensity_df <- sqldf("SELECT *, ROW_NUMBER() OVER() AS row_num
                      FROM exercise_df")
# This query selects only those rows that have matching row number to the row_nums list
intensity_df <- sqldf("SELECT *
                      FROM intensity_df
                      WHERE row_num IN (SELECT * FROM row_nums)")

# Note: An error was raised when I ran them both in 1 query, so I separated them into 2
```

If we plot the data now, the graphs are hard to read since the data is unordered. To fix this, we sort the data first by activity type and then by increasing intensity within each activity type.

```
intensity_df <- intensity_df[
  with(intensity_df, order(ActivityType, Calories.per.kg)),
]
row.names(intensity_df) <- 1:nrow(intensity_df)
intensity_df$Activity <- factor(intensity_df$Activity, levels = intensity_df$Activity) # Changing varia
```

Now that the data is sorted, we plot the data to visualize the relationship between activity intensity and caloric expenditure



From the plot we see that for each activity, caloric expenditure increases monotonically with increasing intensity level. We also see large increases in caloric expenditure with increasing intensity for each activity, sometimes up to a 2- to 3-fold increase from low intensity to high intensity (eg. Cycling).

Next I want to calculate the average difference between low and high intensity activity across each of the activities in order to get an estimate for the magnitude of the effect of exercise intensity on caloric expenditure.

```
# Load the modified data set that contains the 'ActivityType' for each activity (eg. both 'Calisthenics
setwd('C:\\Users\\micha\\Desktop\\Coding Projects (older)\\RStudio Scripts and Data\\Google Capstone Ana
exercise_df_modified <- read.csv('exercise_intensity.csv')

len <- length(exercise_df_modified$ActivityType)

# Define our starting values for the ActivityType and low caloric expenditure. These are updated in the
prev_activity <- exercise_df_modified$ActivityType[1]
low_cal <- exercise_df_modified$Calories.per.kg[1]

# Define the vector that contains all the values of the difference between caloric expenditure for low
diff_vector <- c()
low_vector <- c(low_cal)
high_vector <- c()

# Iterate through the data frame and calculate the difference between low and high intensities for each
for (i in 1:len) {
  current_activity <- exercise_df_modified$ActivityType[i]

  # Each time the activity changes, record the difference between the low and high calorie values and u
  if (current_activity != prev_activity) {
    prev_activity <- current_activity
    high_cal <- exercise_df_modified$Calories.per.kg[i-1] # Save the caloric expenditure of the highest
    diff <- high_cal - low_cal
    diff_vector <- append(diff_vector, diff)
    low_cal <- exercise_df_modified$Calories.per.kg[i] # Save the low caloric expenditure value for the
    low_vector <- append(low_vector, low_cal)
    high_vector <- append(high_vector, high_cal)
  }

  # If at the end of the dataframe, calculate the last value for the difference between high and low in
  else if (i == len) {
    high_cal <- exercise_df_modified$Calories.per.kg[len]
    diff = high_cal - low_cal
    diff_vector <- append(diff_vector, diff)
    high_vector <- append(high_vector, high_cal)
  }
}
cat("The average caloric expenditure of low intensity activities is", mean(low_vector), "calories per kg
```

```
## The average caloric expenditure of low intensity activities is 1.007915 calories per kg bodyweight
```

```
cat("The average caloric expenditure of high intensity activities is", mean(high_vector), "calories per
```

```
## The average caloric expenditure of high intensity activities is 2.059301 calories per kg bodyweight
```

```r
cat("This results in an average difference between low and high intensity activities of ", mean(diff_ve
```

```
## This results in an average difference between low and high intensity activities of  1.051386
```

The calculated difference between low and high intensity levels is 1.05 cal/kg, which is nearly a 2-fold increase from the average low intensity caloric expenditure of 1.01 cal/kg. This data set is experimental, so to investigate the effect of activity intensity level on caloric expenditure, we will next run a t-test and calculate an effect size for our data. This is a repeated-measures experiment, so we will use a paired t-test.

```r
t.test(low_vector, high_vector, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  low_vector and high_vector
## t = -6.5498, df = 18, p-value = 0.000003729
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -1.3886298 -0.7141415
## sample estimates:
## mean difference
##       -1.051386
```

```r
library(lsr) # Load the package to run the 'cohensD()' function
cat("The Cohen's D effect size is", cohensD(low_vector, high_vector, method = 'paired'))
```

```
## The Cohen's D effect size is 1.502626
```

The p-value of 3.7E-6 and effect size of 1.5 shows that the relationship is both statistically significant and strong. We will further discuss our findings and how they inform our recommendations for the company's marketing strategy in Phase 6 (Act) section of this report.

# PHASE 5 - SHARE

In addition to the plots displayed throughout this report of the important trends we identified in the data, we include links to Tableau visualizations below:

## 5.1) Exercise Intensity Box Plot

https://public.tableau.com/app/profile/michael1659/viz/ExerciseIntensityvsCaloriesBurned/Sheet2?publish=yes

## 5.2) Exercise Intensity Bar Chart

https://public.tableau.com/app/profile/michael1659/viz/ExerciseIntensityvsCaloriesBurned2/Sheet2

# PHASE 6 - ACT

In this phase, we consolidate our findings and use them to help inform our company's marketing decisions.

First we revisit the questions from phase 1 (Ask): The questions are reproduced below:

1.1) What are the trends in device usage?
1.2) What are potential use-cases for similar products?
1.3) How could these trends and use-cases apply to our customers?
1.4) How could these insights influence our marketing strategy?

For each of the 3 main insights we gathered from the data (shown in sections 4.1, 4.2, and 4.3 respectively), we will discuss how each of the questions (1.1 - 1.4) pertain to the insight. Section 6 is divided into 3 sub-sections, each corresponding to one of the main insights from section 4.

## 6.1) Heart-Rate Data

**6.1.1) What are the Trends in Device Usage?** In section 4.1) we explored the time-series heart rate data collected by the Fitbit devices and found that they recorded several data points where the user's heart rate was either concerningly low or high. The data could also be plotted in a time-series plot to show fluctuations in heart-rate overtime for users, and indeed, large fluctuations can be observed from these plots. This reveals our first trend in the data, that users tend to have occurrences of very low or very high heart rates.

**6.1.2) What Are Potential Use-Cases for Similar Products?** We suggest that our fit-tracker device could potentially be used to identify when users have a dangerously low or high heart rate and alert them (or even alert medical professionals). Using the device in this way could potentially prevent serious medical incidents from occurring in extreme cases. The device could also be used to monitor users' heart rate data and make sure it is within healthy ranges.

**6.1.3) How Could These Trends and Use-Cases Apply to our Customers?** This use-case could be particularly beneficial for individuals with pre-existing conditions that closely monitor their heart health. The ability to track heart rate data may also be desirable for those interested in general health-tracking.

**6.1.4) How Could These Insights Influence our Marketing Strategy?** We could use these insights to market the device as a tool to track heart health overtime and help to catch potential medical emergencies before the user would otherwise we aware of them. We could us this information to target populations that are interested in heart health tracking. For marketing with these populations, it may also be helpful to point out the other health-tracking features of the device, such as tracking steps, calories burned, and activity intensity. In this way, users could do more comprehensive health-tracking with the device. For example, a user could track their immediate heart health, as well as the total number of steps they walk each day.

## 6.2) The Relationship Between Total Steps and Caloric Expenditure

**6.2.1) What are the Trends in Device Usage?** In section 4.2) we found that there is a strong linear relationship between total steps walked and calories burned both when averaged for each day (across all participants) and when averaged for each participant (across all days). There was a particularly strong relationship between steps walked in a day and calories burned (0.82). This corresponds to an R-squared value of 0.67, meaning that 67% of the variance in calories burned each day could be explained by variance in total steps taken for that day. While our data alone cannot be used to infer causation between steps walked and calories burned, it is well known that physical activity directly leads to caloric expenditure. Therefore, we feel comfortable concluding that in general, increasing steps taken per day will lead to an increase in calories burned.

**6.2.2) What Are Potential Use-Cases for Similar Products?** One use-case is to track the total steps the user takes each day. This could be enhanced with other features such as letting the user select a goal of reaching a target number of steps each day. The user could then choose to have the device send alerts throughout the day to a) let the user know once they have reached the target number of steps for the day, b) notify the user if they are on track to reach their target, or c) remind the user to try to get more steps to reach their goal. It may also be possible to implement a feature where the user inputs into the device their goal-weight and current weight, and the device would then estimate a number of steps to take each day to reach the goal.

**6.2.3) How Could These Trends and Use-Cases Apply to our Customers?** These features could be useful for users by providing concrete and specific daily goals that are personalized for each user. Goals of this nature have been shown to be more likely to be achieved than abstract goals, and could be used to boost user's motivation to increase their activity levels.

**6.2.4) How Could These Insights Influence our Marketing Strategy?** We could market the strong relationship we found between total steps and calories burned in a day to illustrate to users the benefit of tracking steps to achieve weight loss or exercise goals. The additional features mentioned above could also be marketed to users to help them stay motivated and achieve their goals.

## 6.3) Relationship Between Activity Intensity and Caloric Expenditure

**6.3.1) What are the Trends in Device Usage?** In section 4.3) we found moderate correlations (0.18 and 0.26 respectively) between light and moderate activity with calories burned, and a strong correlation (0.61) between vigorous activity and calories burned. We wondered if vigorous activity is particularly effective at increasing caloric expenditure. We chose to further investigate this relationship by finding more data pertaining to activity intensity levels and how they impact caloric expenditure. Ideally, we wanted to find an experimental data set such that we might learn more about the causal relationship between intensity and caloric expenditure (our original data set contains correlational and longitudinal data). We found a dataset that logged data across various intensity levels of 19 distinct activities. We found that all 19 of these activities had caloric expenditure that increased monotonically with increasing intensity - sometimes up to a 3-fold increase from the low intensity version of the activity to the high intensity version. As outlined above in section 4.2), we also found that this relationship was statistically significant ($p = 3.7E\text{-}6$) and has a large effect size of 1.5 when comparing the lowest intensity version to the highest intensity one.

**6.3.2) What Are Potential Use-Cases for Similar Products?** One use case is to track users' intensity levels throughout the day during their activities. Users could then track how much time they have logged in low, medium, and high intensity activity and use this information to adjust their activity levels according to their goals. Additionally, our device could potentially recommend a target high-intensity zone for users to aim for during their exercise. For example a user could start exercise and check their smart device to see that they are in the medium intensity zone, they could then increase their intensity to reach the recommended high intensity zone.

**6.3.3) How Could These Trends and Use-Cases Apply to our Customers?** These use-cases would apply to our customers that are interested in health tracking, those with weight loss related goals, and those with exercise related goals. Our device could help them to effectively track their activity intensity levels and could provide them with tools to help optimize their exercise intensity.

**6.3.4) How Could These Insights Influence our Marketing Strategy?** We could market these functionalities of our device to populations interested in health tracking. We could start by advertising our findings that show that high-intensity activity in particular has a disproportionately large effect on caloric expenditure. From there we could demonstrate how our device can help them track and increase their activity levels to help them reach their specific goals.