Setting up my environment

All datasets were downloaded from Kaggle, as described later in this report. They were hosted locally on my machine via a postgresql server. The following code set up my environment to run the queries, but the code displayed in later cells is only formatted as SQL visually. It is not executable from this notebook at this time.

In [30]:
```r
# Install required packages if they are missing (runs in R)
if (!requireNamespace("DBI", quietly = TRUE)) install.packages("DBI",
if (!requireNamespace("RPostgres", quietly = TRUE)) install.packages("
if (!requireNamespace("dplyr", quietly = TRUE)) install.packages("dply
if (!requireNamespace("dotenv", quietly = TRUE)) {
  install.packages("dotenv", repos = "https://cloud.r-project.org")
}
```

In [31]:
```r
# Load the package
library(dotenv)

# Load a .env file from the notebook working directory (or provide a f
# Returns TRUE on success
loaded <- tryCatch({
  dotenv::load_dot_env(file = ".env")
}, error = function(e) {
  message("Failed to load .env: ", e$message)
  FALSE
})
```

In [32]:
```r
# Load libraries
library(DBI)
library(RPostgres)
library(dplyr)
library(glue)

# Read values from environment
pg_host <- Sys.getenv("PGHOST", unset = "localhost")
pg_port <- as.integer(Sys.getenv("PGPORT", unset = "5432"))
pg_user <- Sys.getenv("PGUSER", unset = "")
pg_password <- Sys.getenv("PGPASSWORD", unset = "")
pg_db <- Sys.getenv("PGDATABASE", unset = "")

# Show current environment values (does NOT print secrets)
cat("PGHOST:   ", Sys.getenv("PGHOST",   unset = "<not set>"), "\n")
cat("PGPORT:   ", Sys.getenv("PGPORT",   unset = "<not set>"), "\n")
cat("PGDATABASE:", Sys.getenv("PGDATABASE", unset = "<not set>"), "\n"
cat("PGUSER:   ", Sys.getenv("PGUSER",   unset = "<not set>"), "\n")
if (nzchar(Sys.getenv("PGPASSWORD"))) {
```

```
  cat("PGPASSWORD: (loaded, not printed)\n")
} else {
  cat("PGPASSWORD: (not set)\n")
}
```

```
PGHOST:     localhost
PGPORT:     5435
PGDATABASE: erindamm
PGUSER:     erindamm
PGPASSWORD: (loaded, not printed)
```

In [33]:
```
# Connect to PostgreSQL
con <- dbConnect(
  RPostgres::Postgres(),
  host = pg_host,
  port = pg_port,
  dbname = pg_db,
  user = pg_user,
  password = pg_password
)

cat("Connection established.\n")

# List tables (shows first few if many)
tables <- dbListTables(con)
cat("Tables (first 20):\n")
print(head(tables, 20))
```

```
Connection established.
Tables (first 20):
 [1] "fitbase_daily_activity_merged"
 [2] "fitbase_daily_calories_merged"
 [3] "fitbase_daily_intensities_merged"
 [4] "fitbase_daily_sleep_merged"
 [5] "fitbase_daily_steps_merged"
 [6] "fitbase_heartrate_seconds_merged"
 [7] "fitbase_hourly_calories_merged"
 [8] "fitbase_hourly_intensities_merged"
 [9] "fitbase_hourly_steps_merged"
[10] "fitbase_minute_calories_narrow_merged"
[11] "fitbase_minute_calories_wide_merged"
[12] "fitbase_minute_intensities_narrow_merged"
[13] "fitbase_minute_intensities_wide_merged"
[14] "fitbase_minute_mets_narrow_merged"
[15] "fitbase_minute_sleep_merged"
[16] "fitbase_minute_steps_narrow_merged"
[17] "fitbase_minute_steps_wide_merged"
[18] "fitbase_weight_log_info_merged"
[19] "health_fitness"
[20] "survey_605"
```

The Business Problem

Bellabeat is looking to analyze existing data in smart device usage to understand how potential users are tracking diferent health related metrics in their daily lives. These trends could be used to identify features that current Bellabeat customers may not know our devices have. These trends could also influence marketing strategy in identifying other Bellabeat products to recommend to existing customers, and which features to highlight in future marketing campaigns.

In this analysis we are looking to identify the following trends:

- frequency of fitness wearable use
- frequency of tracking fitness activity with wearable technology
- impact of wearable fitness trackers on purchasing decisons
- feature used least consistently
- when users are most/least active (to schedule marketing campaigns)
- attitudes of users towards their wearable technology

The Data

For this analysis the marketing team started with the FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius). This dataset contains the personal fitness information from thirty fitbit users. These users consented to the submission of their data including output for physcial activity, heart rate, and sleep monitoriing. It also includes information about daily physcial activity including steps taken, distance traveled, and heart rate that can be used to explore consumer usage habits. This data set has limitations in that it does not explore all the features available with Bellabeat devices and it does not explore any device types other than fitbit. It also doesn't explore general attitudes towards fitness wearables that could be useful for marketing purposes. For this reason, extra data was needed.

Two additional sets of data were sourced to be evaluated as potential supplements to the existing data.

The FitLife: Heath & Fitness Tracking (CCO: Public Domain, dataset made available through Jija Taheri) dataset was considered and evaluated as a possible data source, but as it is synthetic data, it wasn't captured from actual users. Rather, it is a dataset created specifically for the purposes of data exploration and community learning. This dataset didn't contain any data trends that could be applicable to this business question we are attempting to answer in this case study, but was a good exercise in cleaning and analyzing data using various tools, including BigQuery, spreadsheets, and Tableau Public. My changelog for this

dataset can be found here

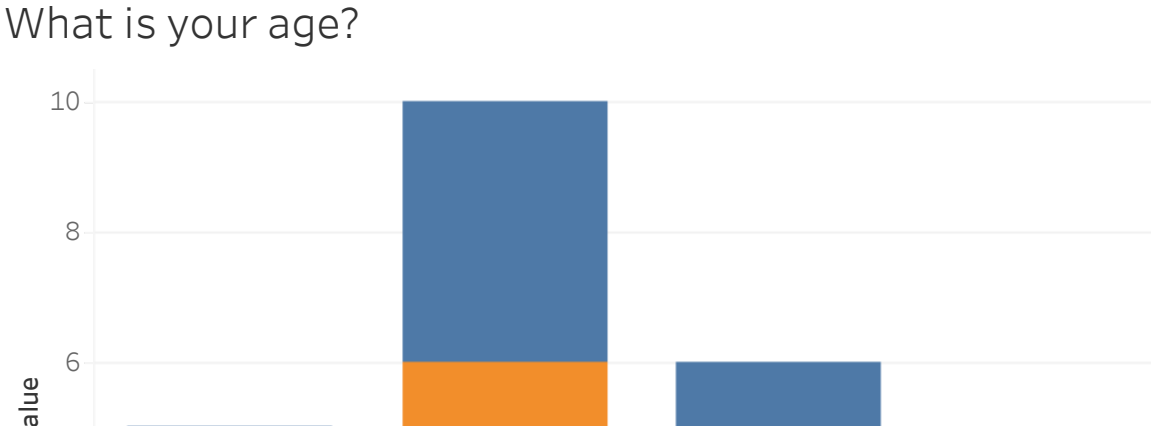
The Fitness Consumer Survey Data dataset (CCO: Public Domain, dataset made available through Harshita Aswani) was also considered and evaluated as a possible data source. This dataset contains survey responses from a variety of respondents about their attitudes and experiences using a fitness wearable. The data was collected from an online survey and all respondents consented to their anonymous responses being shared. The data is primary in nature and appears credible and relatively current. This dataset will be useful in providing context to the Bellabeat executives on broader views of fitness wearables. My changelog for this dataset can be found here

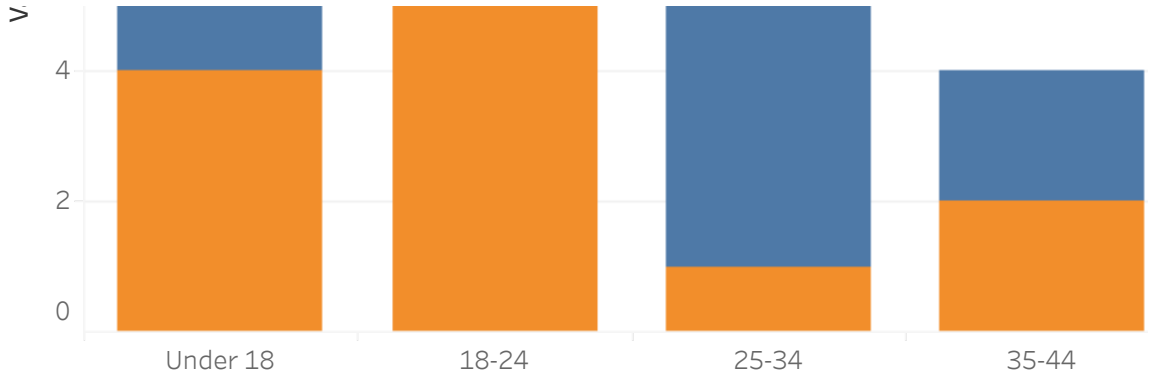
The Analysis

To discuss trends that can apply to Bellabeat customers, we must first understand who those customers are. Using the Fitness Consumer Survey Data, we can see that our customers targeted by this study span ages ranging from under 18 to 64 years old, with most respondents falling in the 18-24 and 25-34 ranges. To better represent our target market segment, we then filter our data to exclude those that self-identified as "Male." This shows that the 18-24 and 25-34 categories still hold as the top two, with the 45-54 category coming in with the third most respondents.

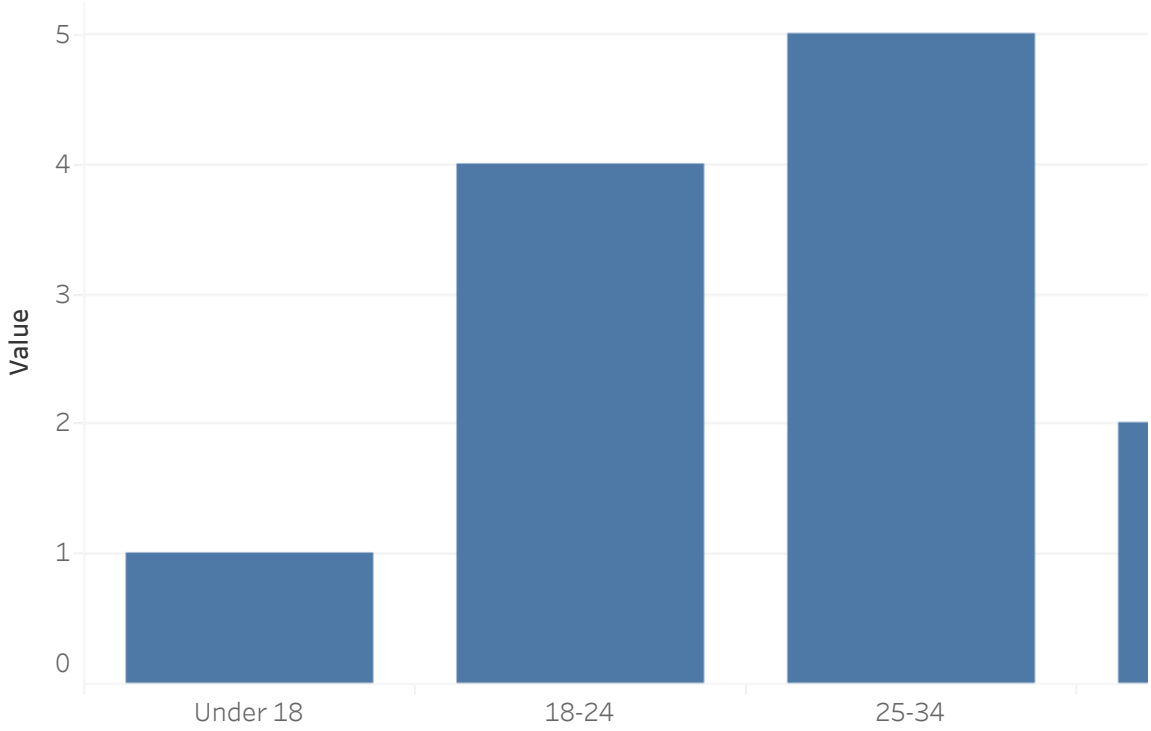
In [34]:
```
html_code <- "
<div class='tableauPlaceholder' id='viz1768615193334' style='position:
"

display_html(html_code)
```





What is your age? (Filtered)



If the embedded image above does not load, please use this link to view it directly on Tableau Public. Our customers span a wide range of occupations and education levels. These images will be useful to refer back to when tailoring campaigns to specific slices of the market, though they provide more general categorizations rather than specific occupational data.

When examining how engaged users feel with their fitness wearable, we see

interesting trends when we filter our graph by age groups. The 18-24 year olds span from negative to very positive responses regarding engagement, while other age groups tend to answer in neighboring pairs like neutral and somewhat engaged, or somewhat and very engaged.

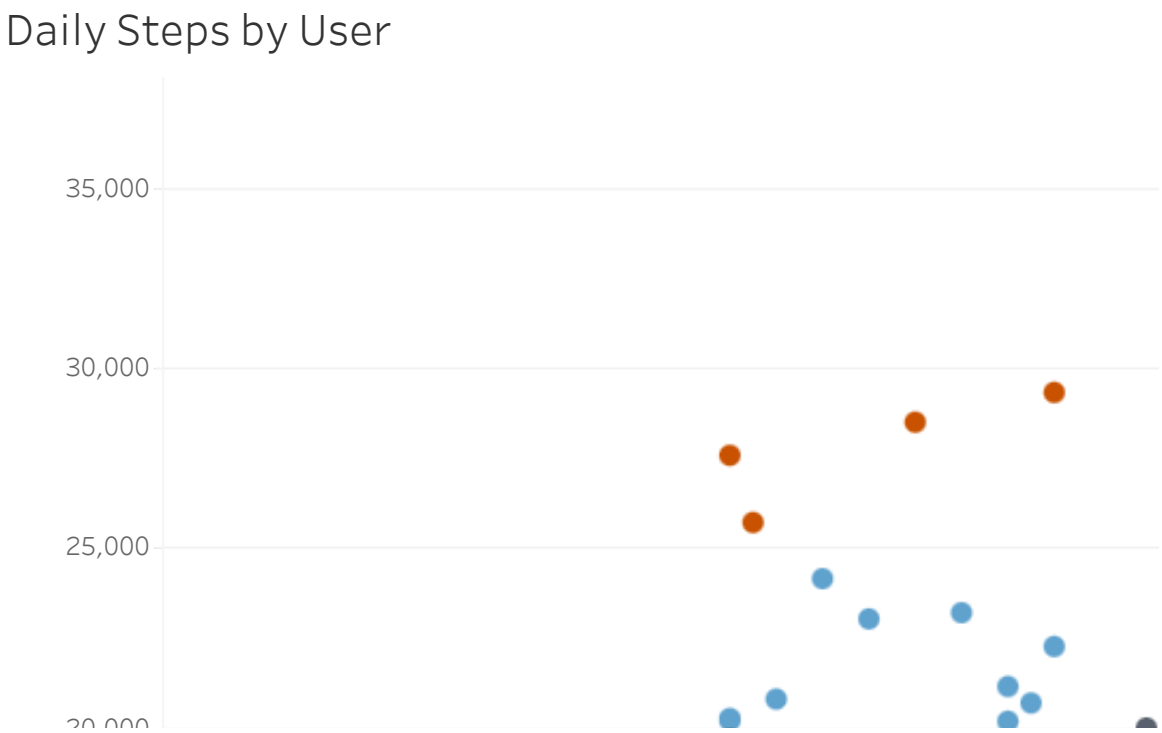
On the next slide we start to examine the different ways users engage with their fitness wearables. Most customers reported a positive impact on their fitness routines, an acceleration in the achievement of their fitness goals, and an increase in both enjoying excercising and mainting motivation to exercise.

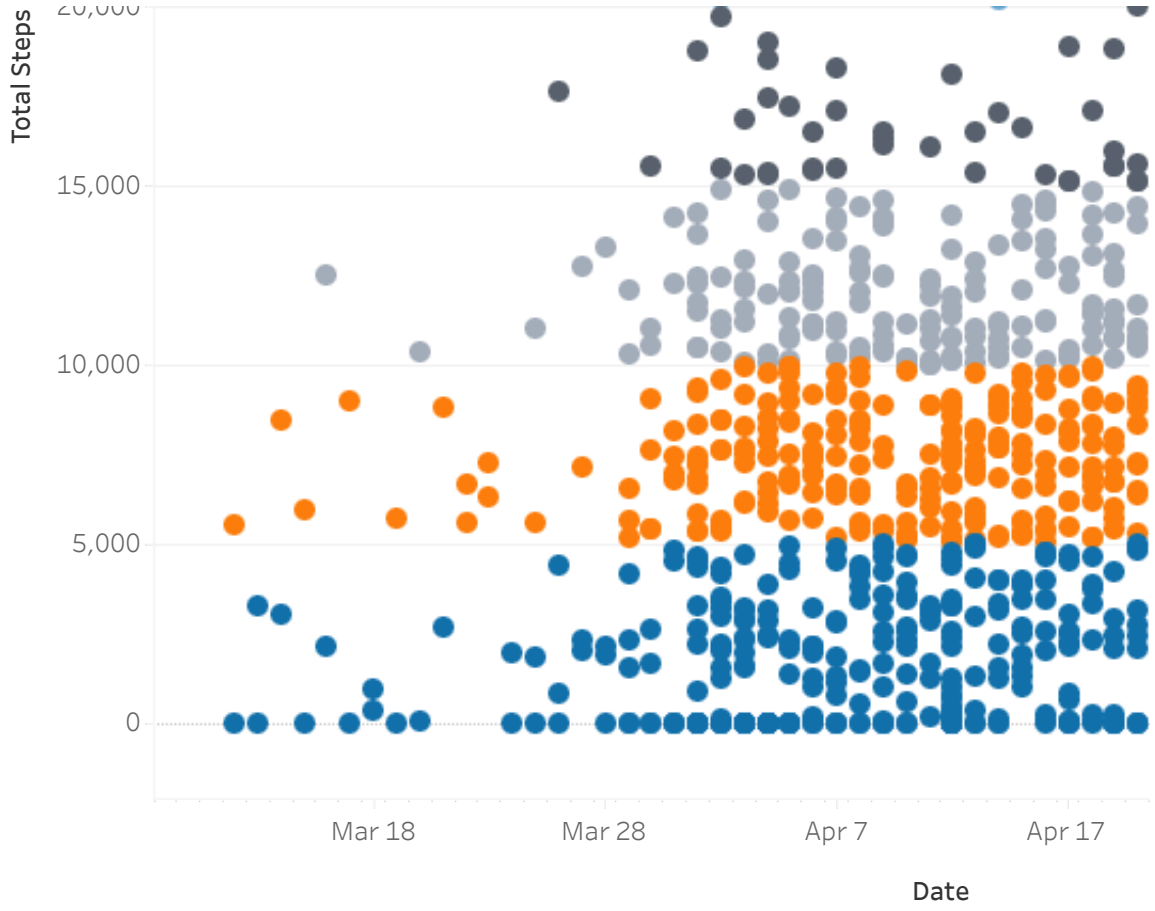
Continuing our exploration of how users engage with their wearables, we see that users give generally favorable responses in seeing an improvement in sleep patterns, in feeling connected to the fitness community through their wearables, and in impact on both their overall health and well-being.

Taking a quick dive into some actual anonymized fitness wearable data, we can spot some trends in how customers use their devices.

In [35]:
```
html_code <- "
<div class='tableauPlaceholder' id='viz1768611038700' style='position:
"

display_html(html_code)
```





If the embedded image above does not load, please use this link to view it directly on Tableau Public.

This first slide gives us a look at the metric that is the most basic foundation of the fitness wearable. The pedometer data. There isn't anything too surprising here. The largest bucket is our lowest step count for the day, encompassing everything below 5000 steps. Second largest subset is the 5000-9999 bucket, with the rest of the buckets having an inverse relationship as the number of steps grows.

When it comes to calories burned per day, our biggest bucket here was acutally the 2000-2999 bucket with the preceeding bucket (1000-1999) with the second largets population. This makes sense as Americans are advised 2000 calories per day is the recommended diet.

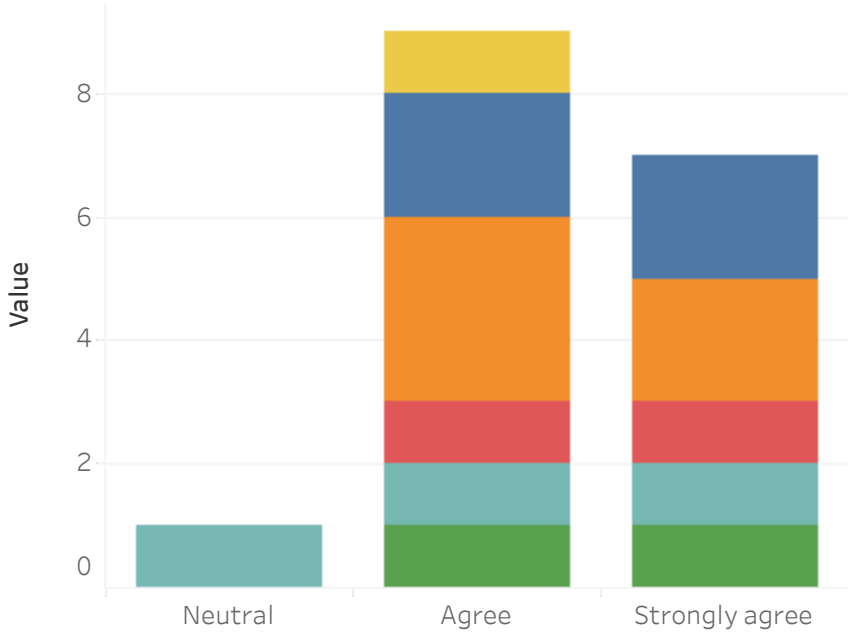
The active minutes broken out by intensity is helpful for taking a look at how people are exercising, vs spending time in a sedentary repose. Again, nothing too surprising here. Most users spend the most time sedentary, which tracks with daily life activities like jobs, commuting, and so on.

The last slide of this deck compares the amount of time user's wearables tracked them as "asleep" vs just "in bed." We can see where some users had very leisurly days with several recording roughly 16 hours in bed with the associated sleep data equalling roughly 12-13 hours of time. The more sleep deprived of these users reported roughly 1 hour of both time in bed and time asleep.

In [36]:
```
html_code <- "
<div class='tableauPlaceholder' id='viz1768615122368' style='position:
"

display_html(html_code)
```

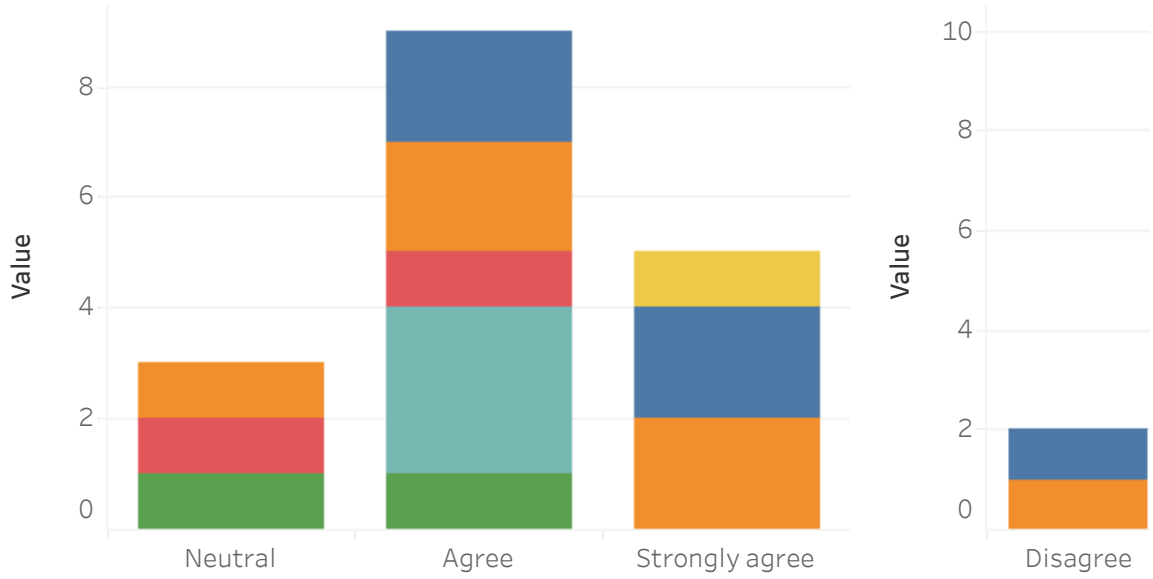
Has using a fitness wearable influenced your decision? (To change your diet?)



Has using a fit your decision?

Has using a fitness wearable influenced your decision? (To join a gym or fitness class?)

Has using a fit your decision? related produ



If the embedded image above does not load, please use this link to view it directly on Tableau Public.

So, now we know a bit about the users, and a bit about what general things they are tracking in their busy lives, let's take a look at their decision making trends. As you can see users reported that using a wearable fitness device influenced them to make other health & fitness changes, such as dietary changes, increasing their exercise activity, joining a gym or fitness class, and purchasing other fitness related products. This is helpful because we can use the information to help us decide which features to highlight in ad campaigns and help target possible repeat customers for new product launches.

Our last slide just enlarges the fourth graph from the previous slide because it is important. The majority of respondents reported that using a fitness wearable influenced their decision to purchase other fitness-related products. This shows that there is a potential repeat purchase audience available, which could lead to possible future products and potential partnerships for limited edition jewelry accessories, etc.

The Recommendations

1. With Bellabeat's current product catalog, it makes sense that our current target demographic is women of childbearing age. Continue to market towards these women while tailoring campaigns towards specific market segments based on topics of interest, favored features, less well-known features, and features that might integrate with different occupations.

2. Look at how our products can be used to track symptoms for women experiencing perimenopause and menopause. As our trackers and algorithms already work to predict fertility, those same features might be usable in an additional life stage.

3. Those that are already engaged with their fitness wearable will likely make additional fitness-related purchases. This could be additional supscription content related to healthy eating, etc.

Changelogs

Fitbase Changelog

This dataset is broken up into two one-month chunks. The first month contains 11 tables covering metrics like steps, calories and sleep broken down over different granularities of time. The second month contains continuatios of these same metrics, but adds 7 additional tables to show daily composites of the metrics as well as wide formats to compliment the narrow ones provided where data is granulated down to the minute.

The tables were located onto my local machine's postgresql server using the following queries to create the tables, plus the data import wizard to fill them in. The 11 tables that were repeated across both months had both files combined into the same table.

```
CREATE TABLE fitbase_daily_activity_merged (
    participant_id BIGINT,
    activity_date DATE,
    total_steps INTEGER,
    total_distance NUMERIC,
    tracker_distance NUMERIC,
    logged_activities_distance NUMERIC,
    very_active_distance NUMERIC,
```

```
    moderately_active_distance NUMERIC,
    light_active_distance NUMERIC,
    sedentary_active_distance NUMERIC,
    very_active_minutes INTEGER,
    fairly_active_minutes INTEGER,
    lightly_active_minutes INTEGER,
    sedentary_minutes INTEGER,
    calories INTEGER)

CREATE TABLE fibase_daily_calories_merged (
    participant_id BIGINT,
    activity_day DATE,
    calories INTEGER)

CREATE TABLE fitbase_daily_intensities_merged (
    participant_id BIGINT,
    activity_day DATE,
    sedentary_minutes INTEGER,
    lightly_active_minutes INTEGER,
    fairly_active_minutes INTEGER,
    very_active_minutes INTEGER,
    sedentary_active_distance NUMERIC,
    light_active_distance NUMERIC,
    moderately_active_distance NUMERIC,
    very_active_distance NUMERIC)

CREATE TABLE fitbase_daily_sleep_merged (
    participant_id BIGINT,
    sleep_day DATE,
    total_sleep_records INTEGER,
    total_minutes_asleep INTEGER,
    total_time_in_bed INTEGER)

CREATE TABLE fitbase_daily_steps_merged (
    participant_id BIGINT,
    activity_day DATE,
    step_total INTEGER)

CREATE TABLE fitbase_heartrate_seconds_merged (
    participant_id BIGINT,
    log_time TIMESTAMP,
    heartrate_value INTEGER
```)

```
CREATE TABLE fitbase_hourly_calories_merged (
    participant_id BIGINT,
    activity_hour TIMESTAMP,
    calories INTEGER)

CREATE TABLE fitbase_hourly_intensities_merged (
```

```
    participant_id BIGINT,
    activity_hour TIMESTAMP,
    total_intensity INTEGER,
    average_intensity NUMERIC)

CREATE TABLE fitbase_hourly_steps_merged ()
    participant_id BIGINT,
    activity_hour TIMESTAMP,
    step_total INTEGER)

CREATE TABLE fitbase_minute_calories_narrow_merged (
    participant_id BIGINT,
    activity_minute TIMESTAMP,
    calories NUMERIC)

CREATE TABLE fitbase_minute_calories_wide_merged (
   participant_id BIGINT,
    activity_hour TIMESTAMP,
    calories00 NUMERIC,
    calories01 NUMERIC,
    calories02 NUMERIC,
    calories03 NUMERIC,
    calories04 NUMERIC,
    calories05 NUMERIC,
    calories06 NUMERIC,
    calories07 NUMERIC,
    calories08 NUMERIC,
    calories09 NUMERIC,
    calories10 NUMERIC,
    calories11 NUMERIC,
    calories12 NUMERIC,
    calories13 NUMERIC,
    calories14 NUMERIC,
    calories15 NUMERIC,
    calories16 NUMERIC,
    calories17 NUMERIC,
    calories18 NUMERIC,
    calories18 NUMERIC,
    calories19 NUMERIC,
    calories20 NUMERIC,
    calories21 NUMERIC,
    calories22 NUMERIC,
    calories23 NUMERIC,
    calories24 NUMERIC,
    calories25 NUMERIC,
    calories26 NUMERIC,
    calories27 NUMERIC,
    calories28 NUMERIC,
    calories29 NUMERIC,
    calories30 NUMERIC,
```

```
    claories31 NUMERIC,
    calories32 NUMERIC,
    calories33 NUMERIC,
    calories34 NUMERIC,
    calories35 NUMERIC,
    calories36 NUMERIC,
    calories37 NUMERIC,
    calories38 NUMERIC,
    calories39 NUMERIC,
    calories40 NUMERIC,
    calories41 NUMERIC,
    calories42 NUMERIC,
    calories43 NUMERIC,
    calories44 NUMERIC,
    calories45 NUMERIC,
    calories46 NUMERIC,
    calories47 NUMERIC,
    calories48 NUMERIC,
    calories49 NUMERIC,
    calories50 NUMERIC,
    calories51 NUMERIC,
    calories52 NUMERIC,
    calories53 NUMERIC,
    calories54 NUMERIC,
    calories55 NUMERIC,
    calories56 NUMERIC,
    calories57 NUMERIC,
    calories59 NUMERIC)

CREATE TABLE fitbase_minute_intensities_narrow_merged (
    participant_id BIGINT,
    activity_minute TIMESTAMP,
    intensities INTEGER)

CREATE TABLE fitbase_minute_intensities_wide_merged (
    participant_id BIGINT,
    activity_hour TIMESTAMP,
    intensity00 NUMERIC,
    intensity01 NUMERIC,
    intensity02 NUMERIC,
    intensity03 NUMERIC,
    intensity04 NUMERIC,
    intensity05 NUMERIC,
    intensity06 NUMERIC,
    intensity07 NUMERIC,
    intensity08 NUMERIC,
    intensity09 NUMERIC,
    intensity10 NUMERIC,
    intensity11 NUMERIC,
    intensity12 NUMERIC,
```

```
    intensity13 NUMERIC,
    intensity14 NUMERIC,
    intensity15 NUMERIC,
    intensity16 NUMERIC,
    intensity17 NUMERIC,
    intensity18 NUMERIC,
    intensity19 NUMERIC,
    intensity20 NUMERIC,
    intensity21 NUMERIC,
    intensity22 NUMERIC,
    intensity23 NUMERIC,
    intensity24 NUMERIC,
    intensity25 NUMERIC,
    intensity26 NUMERIC,
    intensity27 NUMERIC,
    intensity28 NUMERIC,
    intensity29 NUMERIC,
    intensity30 NUMERIC,
    intensity31 NUMERIC,
    intensity32 NUMERIC,
    intensity33 NUMERIC,
    intensity34 NUMERIC,
    intensity35 NUMERIC,
    intensity36 NUMERIC,
    intensity37 NUMERIC,
    intensity38 NUMERIC,
    intensity39 NUMERIC,
    intensity40 NUMERIC,
    intensity41 NUMERIC,
    intensity42 NUMERIC,
    intensity43 NUMERIC,
    intensity44 NUMERIC,
    intensity45 NUMERIC,
    intensity46 NUMERIC,
    intensity47 NUMERIC,
    intensity48 NUMERIC,
    intensity49 NUMERIC,
    intensity50 NUMERIC,
    intensity51 NUMERIC,
    intensity52 NUMERIC,
    intensity53 NUMERIC,
    intensity54 NUMERIC,
    intensity55 NUMERIC,
    intensity56 NUMERIC,
    intensity57 NUMERIC,
    intensity58 NUMERIC,
    intensity59 NUMERIC)

CREATE TABLE fibase_minutes_mets_narrow_merged (
    participant_id BIGINT,
```

```
    activity_minute TIMESTAMP,
    mets INTEGER)

CREATE TABLE fitbase_minute_sleep_merged (
    participant_id BIGINT,
    activity_date TIMESTAMP,
    sleep_value INTEGER,
    log_id BIGINT)

CREATE TABLE fitbase_minute_steps_narrow (
    participant_id BIGINT,
    activity_minute TIMESTAMP,
    steps INTEGER)

CREATE TABLE fitbase_minute_steps_wide_merged (
    participant_id BIGINT,
    activity_hour TIMESTAMP,
    steps00 NUMERIC,
    steps01 NUMERIC,
    steps02 NUMERIC,
    steps03 NUMERIC,
    steps04 NUMERIC,
    steps05 NUMERIC,
    steps06 NUMERIC,
    steps07 NUMERIC,
    steps08 NUMERIC,
    steps09 NUMERIC,
    steps10 NUMERIC,
    steps11 NUMERIC,
    steps12 NUMERIC,
    steps13 NUMERIC,
    steps14 NUMERIC,
    steps15 NUMERIC,
    steps16 NUMERIC,
    steps17 NUMERIC,
    steps18 NUMERIC,
    steps19 NUMERIC,
    steps20 NUMERIC,
    steps21 NUMERIC,
    steps22 NUMERIC,
    steps23 NUMERIC,
    steps24 NUMERIC,
    steps25 NUMERIC,
    steps26 NUMERIC,
    steps27 NUMERIC,
    steps28 NUMERIC,
    steps29 NUMERIC,
    steps30 NUMERIC,
    steps31 NUMERIC,
    steps32 NUMERIC,
```

```
    steps33 NUMERIC,
    steps34 NUMERIC,
    steps35 NUMERIC,
    steps36 NUMERIC,
    steps37 NUMERIC,
    steps38 NUMERIC,
    steps39 NUMERIC,
    steps40 NUMERIC,
    steps41 NUMERIC,
    steps42 NUMERIC,
    steps43 NUMERIC,
    steps44 NUMERIC,
    steps45 NUMERIC,
    steps46 NUMERIC,
    steps47 NUMERIC,
    steps48 NUMERIC,
    steps49 NUMERIC,
    steps50 NUMERIC,
    steps51 NUMERIC,
    steps52 NUMERIC,
    steps53 NUMERIC,
    steps54 NUMERIC,
    steps55 NUMERIC,
    steps56 NUMERIC,
    steps57 NUMERIC,
    steps58 NUMERIC,
    steps59 NUMERIC)

CREATE TABLE fitbase_weight_log_info_merged (
    participant_id BIGINT,
    log_date TIMESTAMP,
    weight_kg NUMERIC,
    weight_pounds NUMERIC,
    fat NUMERIC,
    bmi NUMERIC,
    is_manual_report TEXT,
    log_id BIGINT)
```

1. Investigating NULL Values

The following queries were run to investigate any possible NULL values in the dataset.

```
SELECT *
FROM fitbase_daily_activity_merged
WHERE participant_id IS NULL
    OR activity_date IS NULL
    OR total_steps IS NULL
```

```
    OR total_distance IS NULL
    OR tracker_distance IS NULL
    OR logged_activities_distance IS NULL
    OR very_active_distance IS NULL
    OR moderately_active_distance IS NULL
    OR light_active_distance IS NULL
    OR sedentary_active_distance IS NULL
    OR very_active_minutes IS NULL
    OR fairly_active_minutes IS NULL
    OR lightly_active_minutes IS NULL
    OR sedentary_minutes IS NULL
    OR calories IS NULL;

SELECT *
FROM fitbase_daily_calories_merged
WHERE participant_id IS NULL
    OR activity_day IS NULL
    OR calories IS NULL;

SELECT *
FROM fitbase_daily_intensities_merged
WHERE participant_id IS NULL
    OR activity_day IS NULL
    OR sedentary_minutes IS NULL
    OR lightly_active_minutes IS NULL
    OR fairly_active_minutes IS NULL
    OR very_active_minutes IS NULL
    OR sedentary_active_distance IS NULL
    OR light_active_distance IS NULL
    OR moderately_active_distance IS NULL
    OR very_active_distance IS NULL;

SELECT *
FROM fitbase_daily_sleep_merged
WHERE participant_id IS NULL
    OR sleep_day IS NULL
    OR total_sleep_records IS NULL
    OR total_minutes_asleep IS NULL
    OR total_time_in_bed IS NULL;

SELECT *
FROM fitbase_daily_steps_merged
WHERE participant_id IS NULL
    OR activity_day IS NULL
    OR step_total IS NULL;

SELECT *
FROM fitbase_heartrate_seconds_merged
WHERE participant_id IS NULL
    OR log_time IS NULL
```

```
    OR heartrate_value IS NULL;

SELECT *
FROM fitbase_hourly_calories_merged
WHERE participant_id IS NULL
    OR activity_hour IS NULL
    OR calories IS NULL;

SELECT *
FROM fitbase_hourly_intensities_merged
WHERE participant_id IS NULL
    OR activity_hour IS NULL
    OR total_intensity IS NULL
    OR average_intensity IS NULL;

SELECT *
FROM fitbase_hourly_steps_merged
WHERE participant_id IS NULL
    OR activity_hour IS NULL
    OR step_total IS NULL;

SELECT *
FROM fitbase_minute_calories_narrow_merged
WHERE participant_id IS NULL
    OR activity_minute IS NULL
    OR calories IS NULL;

SELECT *
FROM fitbase_minute_calories_wide_merged
WHERE
          participant_id IS NULL
       OR activity_hour IS NULL
       OR calories00 IS NULL
    OR calories01 IS NULL
    OR calories02 IS NULL
    OR calories03 IS NULL
    OR calories04 IS NULL
    OR calories05 IS NULL
    OR calories06 IS NULL
    OR calories07 IS NULL
    OR calories08 IS NULL
    OR calories09 IS NULL
    OR calories10 IS NULL
    OR calories11 IS NULL
    OR calories12 IS NULL
    OR calories13 IS NULL
    OR calories14 IS NULL
    OR calories15 IS NULL
    OR calories16 IS NULL
    OR calories17 IS NULL
```

```
    OR calories18 IS NULL
    OR calories19 IS NULL
    OR calories20 IS NULL
    OR calories21 IS NULL
    OR calories22 IS NULL
    OR calories23 IS NULL
    OR calories24 IS NULL
    OR calories25 IS NULL
    OR calories26 IS NULL
    OR calories27 IS NULL
    OR calories28 IS NULL
    OR calories29 IS NULL
    OR calories30 IS NULL
    OR calories31 IS NULL
    OR calories32 IS NULL
    OR calories33 IS NULL
    OR calories34 IS NULL
    OR calories35 IS NULL
    OR calories36 IS NULL
    OR calories37 IS NULL
    OR calories38 IS NULL
    OR calories39 IS NULL
    OR calories40 IS NULL
    OR calories41 IS NULL
    OR calories42 IS NULL
    OR calories43 IS NULL
    OR calories44 IS NULL
    OR calories45 IS NULL
    OR calories46 IS NULL
    OR calories47 IS NULL
    OR calories48 IS NULL
    OR calories49 IS NULL
    OR calories50 IS NULL
    OR calories51 IS NULL
    OR calories52 IS NULL
    OR calories53 IS NULL
    OR calories54 IS NULL
    OR calories55 IS NULL
    OR calories56 IS NULL
    OR calories57 IS NULL
    OR calories58 IS NULL
    OR calories59 IS NULL;

SELECT *
FROM fitbase_minute_intensities_narrow_merged
WHERE
          participant_id IS NULL
       OR activity_minute IS NULL
       OR intensities IS NULL;
```

```sql
SELECT *
FROM fitbase_minute_steps_wide_merged
WHERE
          participant_id IS NULL
       OR activity_hour IS NULL
       OR intensities00 IS NULL
    OR intensities01 IS NULL
    OR intensities02 IS NULL
    OR intensities03 IS NULL
    OR intensities04 IS NULL
    OR intensities05 IS NULL
    OR intensities06 IS NULL
    OR intensities07 IS NULL
    OR intensities08 IS NULL
    OR intensities09 IS NULL
    OR intensities10 IS NULL
    OR intensities11 IS NULL
    OR intensities12 IS NULL
    OR intensities13 IS NULL
    OR intensities14 IS NULL
    OR intensities15 IS NULL
    OR intensities16 IS NULL
    OR intensities17 IS NULL
    OR intensities18 IS NULL
    OR intensities19 IS NULL
    OR intensities20 IS NULL
    OR intensities21 IS NULL
    OR intensities22 IS NULL
    OR intensities23 IS NULL
    OR intensities24 IS NULL
    OR intensities25 IS NULL
    OR intensities26 IS NULL
    OR intensities27 IS NULL
    OR intensities28 IS NULL
    OR intensities29 IS NULL
    OR intensities30 IS NULL
    OR intensities31 IS NULL
    OR intensities32 IS NULL
    OR intensities33 IS NULL
    OR intensities34 IS NULL
    OR intensities35 IS NULL
    OR intensities36 IS NULL
    OR intensities37 IS NULL
    OR intensities38 IS NULL
    OR intensities39 IS NULL
    OR intensities40 IS NULL
    OR intensities41 IS NULL
    OR intensities42 IS NULL
    OR intensities43 IS NULL
    OR intensities44 IS NULL
```

```
    OR intensities45 IS NULL
    OR intensities46 IS NULL
    OR intensities47 IS NULL
    OR intensities48 IS NULL
    OR intensities49 IS NULL
    OR intensities50 IS NULL
    OR intensities51 IS NULL
    OR intensities52 IS NULL
    OR intensities53 IS NULL
    OR intensities54 IS NULL
    OR intensities55 IS NULL
    OR intensities56 IS NULL
    OR intensities57 IS NULL
    OR intensities58 IS NULL
    OR intensities59 IS NULL;

SELECT *
FROM fitbase_minute_mets_narrow_merged
WHERE
          participant_id IS NULL
       OR activity_minute IS NULL
       OR mets IS NULL;

SELECT *
FROM fitbase_minute_sleep_merged
WHERE
          participant_id IS NULL
       OR activity_date IS NULL
       OR sleep_value IS NULL
    OR log_id IS NULL;

SELECT *
FROM fitbase_minute_steps_narrow_merged
WHERE
          participant_id IS NULL
       OR activity_minute IS NULL
       OR steps IS NULL;

SELECT *
FROM fitbase_minute_steps_wide_merged
WHERE
          participant_id IS NULL
       OR activity_hour IS NULL
       OR steps00 IS NULL
    OR steps01 IS NULL
    OR steps02 IS NULL
    OR steps03 IS NULL
    OR steps04 IS NULL
    OR steps05 IS NULL
    OR steps06 IS NULL
```

```
    OR steps07 IS NULL
    OR steps08 IS NULL
    OR steps09 IS NULL
    OR steps10 IS NULL
    OR steps11 IS NULL
    OR steps12 IS NULL
    OR steps13 IS NULL
    OR steps14 IS NULL
    OR steps15 IS NULL
    OR steps16 IS NULL
    OR steps17 IS NULL
    OR steps18 IS NULL
    OR steps19 IS NULL
    OR steps20 IS NULL
    OR steps21 IS NULL
    OR steps22 IS NULL
    OR steps23 IS NULL
    OR steps24 IS NULL
    OR steps25 IS NULL
    OR steps26 IS NULL
    OR steps27 IS NULL
    OR steps28 IS NULL
    OR steps29 IS NULL
    OR steps30 IS NULL
    OR steps31 IS NULL
    OR steps32 IS NULL
    OR steps33 IS NULL
    OR steps34 IS NULL
    OR steps35 IS NULL
    OR steps36 IS NULL
    OR steps37 IS NULL
    OR steps38 IS NULL
    OR steps39 IS NULL
    OR steps40 IS NULL
    OR steps41 IS NULL
    OR steps42 IS NULL
    OR steps43 IS NULL
    OR steps44 IS NULL
    OR steps45 IS NULL
    OR steps46 IS NULL
    OR steps47 IS NULL
    OR steps48 IS NULL
    OR steps49 IS NULL
    OR steps50 IS NULL
    OR steps51 IS NULL
    OR steps52 IS NULL
    OR steps53 IS NULL
    OR steps54 IS NULL
    OR steps55 IS NULL
    OR steps56 IS NULL
```

```
    OR steps57 IS NULL
    OR steps58 IS NULL
    OR steps59 IS NULL;

SELECT *
FROM fitbase_weight_log_info_merged
WHERE
           participant_id IS NULL
        OR log_date IS NULL
        OR weight_kg IS NULL
        OR weight_pounds IS NULL
        OR fat IS NULL
        OR bmi IS NULL
        OR is_manual_report IS NULL
        or log_id IS NULL;
```

This query returned many rows and at quick glance it appears the 'fat' column contains several NULL values. The following script was run to discern if there were any non_NULL values in the column.

```
SELECT *
FROM fitbase_weight_log_info_merged
WHERE
           participant_id IS NULL
        OR log_date IS NULL
        OR weight_kg IS NULL
        OR weight_pounds IS NULL
        OR fat IS NOT NULL
        OR bmi IS NULL
        OR is_manual_report IS NULL
        or log_id IS NULL;
```

There were 4 rows returned with values in the 'fat' column. It was noted that all 4 rows also contained the value 'TRUE' in the 'is_manual_report' column and the following query was run to determine the reporting status of the NULL values in the column.

```
SELECT fat, is_manual_report
FROM fitbase_weight_log_info_merged
WHERE
       fat IS NULL;
```

It is determined that the NULL values are associated with both 'TRUE' and 'FALSE' responses. It is likely that only some participants had models that specifically tracked the 'fat' metric. With only 4 rows in the entire table with non-NULL values, there is not a significant enough sample size to analyze for useful trends among so much other data. This column will be disregarded in further analysis.

The rest of the columns were evaluated individually to determine if there were any other NULL values.

```
SELECT *
FROM fitbase_weight_log_info_merged
WHERE
          participant_id IS NULL;

SELECT *
FROM fitbase_weight_log_info_merged
WHERE
          weight_kg IS NULL;

SELECT *
FROM fitbase_weight_log_info_merged
WHERE
          weight_pounds IS NULL;

SELECT *
FROM fitbase_weight_log_info_merged
WHERE
          fat IS NULL;

SELECT *
FROM fitbase_weight_log_info_merged
WHERE
          bmi IS NULL;

SELECT *
FROM fitbase_weight_log_info_merged
WHERE
          is_manual_report IS NULL;

SELECT *
FROM fitbase_weight_log_info_merged
WHERE
          log_id IS NULL;
```

There were no other NULL values to investigate.

2. Summarizing the Data

The following scripts were run to summarize and analyze the data. A summary table was created in a separate tab in the fitbase_daily_activity_merged_full spreadsheet file.

a. Steps Buckets

```sql
SELECT COUNT(total_steps) AS "0–4999"
FROM fitbase_daily_activity_merged
WHERE total_steps BETWEEN 0 AND 4999;

SELECT COUNT(total_steps) AS "5000–9999"
FROM fitbase_daily_activity_merged
WHERE total_steps BETWEEN 5000 AND 9999;

SELECT COUNT(total_steps) AS "10000–14999"
FROM fitbase_daily_activity_merged
WHERE total_steps BETWEEN 10000 AND 14999;

SELECT COUNT(total_steps) AS "15000–19999"
FROM fitbase_daily_activity_merged
WHERE total_steps BETWEEN 15000 AND 19999;

SELECT COUNT(total_steps) AS "20000–24999"
FROM fitbase_daily_activity_merged
WHERE total_steps BETWEEN 20000 AND 24999;

SELECT COUNT(total_steps) AS "24000–29999"
FROM fitbase_daily_activity_merged
WHERE total_steps BETWEEN 24000 AND 29999;

SELECT COUNT(total_steps) AS "30000–34999"
FROM fitbase_daily_activity_merged
WHERE total_steps BETWEEN 30000 AND 34999;

SELECT COUNT(total_steps) AS "35000+"
FROM fitbase_daily_activity_merged
WHERE total_steps > 35000;
```

b. Calorie Buckets

```sql
SELECT COUNT(calories) AS "0–999"
FROM fitbase_daily_activity_merged
WHERE calories BETWEEN 0 AND 999;

SELECT COUNT(calories) AS "1000–1999"
FROM fitbase_daily_activity_merged
WHERE calories BETWEEN 1000 AND 1999;

SELECT COUNT(calories) AS "2000–2999"
FROM fitbase_daily_activity_merged
WHERE calories BETWEEN 1000 AND 1999;

SELECT COUNT(calories) AS "3000–3999"
```

```
FROM fitbase_daily_activity_merged
WHERE calories BETWEEN 3000 AND 3999;

SELECT COUNT(calories) AS "4000+"
FROM fitbase_daily_activity_merged
WHERE calories > 5000;
```

C. Sleep Buckets

```
SELECT COUNT(total_minutes_asleep) AS "0-250"
FROM fitbase_daily_sleep_merged
WHERE total_minutes_asleep BETWEEN 0 AND 250;

SELECT COUNT(total_minutes_asleep) AS "250-499"
FROM fitbase_daily_sleep_merged
WHERE total_minutes_asleep BETWEEN 250 AND 499;

SELECT COUNT(total_minutes_asleep) AS "500-749"
FROM fitbase_daily_sleep_merged
WHERE total_minutes_asleep BETWEEN 500 AND 749;

SELECT COUNT(total_minutes_asleep) AS "750+"
FROM fitbase_daily_sleep_merged
WHERE total_minutes_asleep > 750;
```

d. Time in Bed Buckets

```
SELECT COUNT(total_time_in_bed) AS "0-250"
FROM fitbase_daily_sleep_merged
WHERE total_time_in_bed BETWEEN 0 AND 250;

SELECT COUNT(total_time_in_bed) AS "250-499"
FROM fitbase_daily_sleep_merged
WHERE total_time_in_bed BETWEEN 250 AND 499;

SELECT COUNT(total_time_in_bed) AS "500-749"
FROM fitbase_daily_sleep_merged
WHERE total_time_in_bed BETWEEN 500 AND 749;

SELECT COUNT(total_time_in_bed) AS "750+"
FROM fitbase_daily_sleep_merged
WHERE total_time_in_bed > 750;
```

3. Visualizing the Data

These new tables were exported as single csv files to be uploaded to Tableau for visual analysis. This data lent itself well to circle plotting, with the data generally

divided up into quarter bins to make it easier to spot visual trends.

In [37]:
```
install.packages("IRdisplay")
```

```
The downloaded binary packages are in
	/var/folders/85/lvfl5s8j30g_2hrj1qv31p780000gn/T//Rtmp7esPB1/do
wnloaded_packages
```

In [38]:
```
library(IRdisplay)

html_code <- "
<div class='tableauPlaceholder' id='viz1768611038700' style='position:
"

display_html(html_code)
```

Daily Steps by User

4. Conclusion

FitLife Dataset Changelog

This dataset is too large to view in a spreadsheet, so this notebook is a changelog for using SQL to clean the health_fitness_dataset. The table was created using the following query, followed by using the import wizard to upload the data from the csv file.

```
CREATE TABLE health_fitness (
        participant_id INTEGER,
        activity_date DATE,
        age INTEGER,
        gender TEXT,
        height_cm NUMERIC,
        weight_kg NUMERIC,
        activity_type TEXT,
        duration_minutes INTEGER,
        intensity TEXT,
        calories_burned NUMERIC,
        avg_heart_rate INTEGER,
        hours_sleep NUMERIC,
```

```
        stress_level INTEGER,
        daily_steps INTEGER,
        hydration_level NUMERIC,
        bmi NUMERIC,
        resting_heart_rate NUMERIC,
        blood_pressure_systolic NUMERIC,
        blood_pressure_diastolic NUMERIC,
        health_condition TEXT,
        smoking_status TEXT,
        fitness_level NUMERIC
)
```

1. Investigating NULL Values

Variations of the following script were run for every column to discover if there were any NULL values.

```
SELECT *
FROM health_fitness
WHERE participant_id IS NULL;
```

As expected from a synthetic dataset, there were no rows containing NULL values that needed to be excluded from the dataset.

2. Summarizing Data

The following queries were run to summarize the data. This summary table was created in Google Sheets to compile these results.

a. Summarizing Gender

```
SELECT
  gender,
  COUNT(DISTINCT participant_id) AS user_count
FROM health_fitness
GROUP BY gender;
```

b. Summarizing Activity Types

First, this query was run to summarize the activity types, split up by intensity:

```
SELECT
  activity_type, intensity,
  COUNT(DISTINCT participant_id) AS activity_type_count,
  COUNT(participant_id) AS activity_session_count,
  SUM(duration_minutes) AS activity_duration_count,
```

```
  ROUND(AVG(duration_minutes), 2) AS
activity_duration_average_in_minutes,
  ROUND(MIN(duration_minutes), 2) AS
activity_duration_min_in_minutes,
  ROUND(MAX(duration_minutes), 2) AS
activity_duration_max_in_minutes
FROM health_fitness
GROUP BY activity_type, intensity
ORDER BY activity_type,
  CASE intensity
        WHEN 'Low' THEN 1
        WHEN 'Medium' THEN 2
        WHEN 'High' THEN 3
        ELSE NULL
        END
```

c. Intensity minutes breakdown

```
SELECT
  intensity,
  COUNT(intensity) AS intensity_session_total_count
FROM health_fitness
GROUP BY
  intensity
ORDER BY
  CASE intensity
        WHEN 'Low' THEN 1
        WHEN 'Medium' THEN 2
        WHEN 'High' THEN 3
        ELSE NULL
        END
```

d. Min, Max, Avg Summaries by User

```
SELECT
  activity_type,
  intensity,
  COUNT(DISTINCT participant_id) AS
distinct_participant_count,
  ROUND(AVG(stress_level), 2) AS stress_average,
  ROUND(MIN(stress_level), 2) AS stress_min,
  ROUND(MAX(stress_level), 2) AS stress_max,
  ROUND(AVG(hours_sleep), 2) AS hours_sleep_average,
  ROUND(MIN(hours_sleep), 2) AS hours_sleep_min,
  ROUND(MAX(hours_sleep), 2) AS hours_sleep_max,
  ROUND(AVG(weight_kg), 2) AS weight_kg_average,
  ROUND(MIN(weight_kg), 2) AS weight_kg_min,
  ROUND(MAX(weight_kg), 2) AS weight_kg_max,
```

```
  ROUND(AVG(hydration_level), 2) AS hydration_level_average,
  ROUND(MIN(hydration_level), 2) AS hydration_level_min,
  ROUND(MAX(hydration_level), 2) AS hydration_level_max,
  ROUND(AVG(daily_steps), 0) AS daily_steps_average,
  ROUND(MIN(daily_steps), 0) AS daily_steps_min,
  ROUND(MAX(daily_steps), 0) AS daily_steps_max,
  ROUND(AVG(age), 2) AS average_age,
  ROUND(AVG(duration_minutes), 0) AS
duration_average_in_minutes,
  ROUND(MIN(duration_minutes), 0) AS duration_min_in_minutes,
  ROUND(MAX(duration_minutes), 0) AS duration_max_in_minutes,
  ROUND(AVG(bmi), 2) AS bmi_average,
  ROUND(MIN(bmi), 2) AS bmi_min,
  ROUND(MAX(bmi), 2) AS bmi_max,
  ROUND(AVG(resting_heart_rate), 1) AS
resting_heart_rate_average,
  ROUND(MIN(resting_heart_rate), 1) AS resting_heart_rate_min,
  ROUND(MAX(resting_heart_rate), 1) AS resting_heart_rate_max,
  ROUND(AVG(blood_pressure_diastolic), 1) AS
blood_pressure_diastolic_average,
  ROUND(MIN(blood_pressure_diastolic), 1) AS
blood_pressure_diastolic_min,
  ROUND(MAX(blood_pressure_diastolic), 1) AS
blood_pressure_diastolic_max,
  ROUND(AVG(blood_pressure_systolic), 1) AS
blood_pressure_systolic_average,
  ROUND(MIN(blood_pressure_systolic), 1) AS
blood_pressure_systolic_min,
  ROUND(MAX(blood_pressure_systolic), 1) AS
blood_pressure_systolic_max,
  ROUND(AVG(calories_burned), 0) AS calories_burned_average,
  ROUND(MIN(calories_burned), 0) AS calories_burned_min,
  ROUND(MAX(calories_burned), 0) AS calories_burned_max
FROM health_fitness
GROUP BY activity_type, intensity
ORDER BY activity_type,
  CASE intensity
    WHEN 'Low' THEN 1
    WHEN 'Medium' THEN 2
    WHEN 'High' THEN 3
    ELSE NULL
  END;
```

e. Min, Max, Avg Summaries by Date

```
SELECT
  DISTINCT date,
  ROUND(AVG(stress_level), 2) AS stress_average,
  ROUND(MIN(stress_level), 2) AS stress_min,
```

```
  ROUND(MAX(stress_level), 2) AS stress_max,
  ROUND(AVG(hours_sleep), 2) AS hours_sleep_average,
  ROUND(MIN(hours_sleep), 2) AS hours_sleep_min,
  ROUND(MAX(hours_sleep), 2) AS hours_sleep_max,
  ROUND(AVG(weight_kg), 2) AS weight_kg_average,
  ROUND(MIN(weight_kg), 2) AS weight_kg_min,
  ROUND(MAX(weight_kg), 2) AS weight_kg_max,
  ROUND(AVG(hydration_level), 2) AS hydration_level_average,
  ROUND(MIN(hydration_level), 2) AS hydration_level_min,
  ROUND(MAX(hydration_level), 2) AS hydration_level_max,
  ROUND(AVG(daily_steps)) AS daily_steps_average,
  ROUND(MIN(daily_steps)) AS daily_steps_min,
  ROUND(MAX(daily_steps)) AS daily_steps_max,
  ROUND(AVG(age), 2) AS average_age,
  ROUND(AVG(duration_minutes)) AS duration_average_in_minutes,
  ROUND(MIN(duration_minutes)) AS duration_min_in_minutes,
  ROUND(MAX(duration_minutes)) AS duration_max_in_minutes,
  ROUND(AVG(bmi)) AS bmi_average,
  ROUND(MIN(bmi)) AS bmi_min,
  ROUND(MAX(bmi)) AS bmi_max,
  ROUND(AVG(resting_heart_rate)) AS
resting_heart_rate_average,
  ROUND(MIN(resting_heart_rate)) AS resting_heart_rate_min,
  ROUND(MAX(resting_heart_rate)) AS resting_heart_rate_max,
  ROUND(AVG(blood_pressure_diastolic)) AS
blood_pressure_diastolic_average,
  ROUND(MIN(blood_pressure_diastolic)) AS
blood_pressure_diastolic_min,
  ROUND(MAX(blood_pressure_diastolic)) AS
blood_pressure_diastolic_max,
  ROUND(AVG(blood_pressure_systolic)) AS
blood_pressure_systolic_average,
  ROUND(MIN(blood_pressure_systolic)) AS
blood_pressure_systolic_min,
  ROUND(MAX(blood_pressure_systolic)) AS
blood_pressure_systolic_max,
  ROUND(AVG(calories_burned)) AS calories_burned_average,
  ROUND(MIN(calories_burned)) AS calories_burned_min,
  ROUND(MAX(calories_burned)) AS calories_burned_max,
  ROUND(AVG(duration_minutes)) AS duration_minutes_average,
  ROUND(MIN(duration_minutes)) AS duration_minutes_min,
  ROUND(MAX(duration_minutes)) AS duration_minutes_max,
FROM health_fitness
GROUP BY date;
```

f. Summarizing While Filtering for Gender

Bellabeat has positioned itself as a health & wellness company for women, so the

data has been filtered to exclude men by adding the following to each summary query written in the previous section to filter the men out of the dataset without altering the dataset itself.

```
WHERE gender <> "M"
```

g. Summing activities by user and date without dividing by intensity

Breaking the sums up created a potential for confusing data as each user could be counted in each intensity category, meaning just summing those values would not be accurate in counting unique users for each activity.

```
SELECT
    activity_type,
    COUNT(DISTINCT participant_id) AS unique_users
FROM health_fitness
WHERE gender <> 'M'
GROUP BY activity_type
ORDER BY activity_type;
```

h. Activity Session Counts

```
SELECT
  activity_type,
  COUNT(activity_type) AS activity_session_count
FROM health_fitness
WHERE gender <> 'M'
GROUP BY
  activity_type
ORDER BY
  activity_type;
```

i. Activity Duration Summary

```
SELECT
  activity_type,
  SUM(duration_minutes) AS activity_duration_in_minutes
FROM health_fitness
WHERE gender <> 'M'
GROUP BY
  activity_type
ORDER BY
  activity_type;
```

3. Data Visualization

The data was visualized using various tables in Tableau Public. This dataset was found to only display trends that were steady and show no growth or decrease. This makes sense as the data is synthetic, and thus wouldn't have any surprising trends to be discovered. These visualiztions were not used in the final report.

4. Conclusion

This dataset is not useful for this case study but was a good chance to practice cleaning and analyzing data using various tools such as BigQuery, spreadsheets, and Tableau Public.

bellabeat fitness consumer survey changelog

This dataset is small enough to view as a spreadsheet as it only contains data from 30 respondents and over 21 questions, plus a timestamp. While the data was examined in a spreadsheet, it was uploaded to the database using SQL queries as a practice exercise.

The table was created using the following query, followed by using the import wizard to upload the data from the csv file.

```
CREATE TABLE survey_605 (
        response_timestamp timestamp,
        age text,
        gender text,
        education_level text,
        occupation text,
        weekly_exercise_frequency text,
        length_wearable_history text,
        wearable_use_frequency text,
        fitness_data_tracking_frequency text,
        impact_fitness_routine text,
        impact_fitness_motivation text,
        impact_exercise_enjoyment text,
        wearable_engagement text,
        community_connection text,
        impact_fitness_goal_achievement text,
        impact_overall_health text,
        impact_sleep_patterns text,
        impact_overall_wellbeing text,
```

```
        influence_exercise_frequency text,
        influence_fitness_purchases text,
        influence_join_gym text,
        influence_dietary_decision text
);
```

1. Investigating NULL Values

There were no null values in the dataset.

2. Summarizing data

A second sheet was created in the survey 605 file as a summary table. As the various responses were tallied, it is evident that the responses were all multiple choice on the survey, though that could not be verified from the dataset's description. This would have been helpful as it is currently unknown if there are any answer options that recieved zero responses, which would change the appearance of data visualizations. As it is, I investigated how many responses of "Strongly disagree" there were in the whole dataset as many of the questions used the same scale ratings as possible answers. There was exactly one "Strongly Disagree" response in the whole sheet, which means there are likely multiple questions with options that recieved zero response.

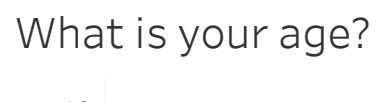
Since all of the answers are standardized, the entire summary table was created using COUNTIF funtions in google sheets.

3. Data Visualization

The data was visualized using various tables in Tableau Public. This data lent itself well to standard bar graphs, but with more granularity, so that consumer opinions can be compared across different questions for the same age group more easily. The viz was split into two stories to facilitate linking data in a functional narrative.

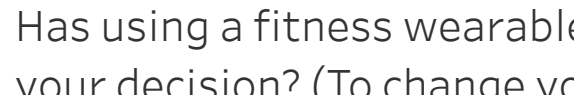
In [39]:
```
html_code <- "
<div class='tableauPlaceholder' id='viz1768615193334' style='position:
"

display_html(html_code)
```



In [40]:
```
html_code <- "
<div class='tableauPlaceholder' id='viz1768615122368' style='position:
"

display_html(html_code)
```



4. Conclusion

This dataset is useful in that it provides demographic information, provides insight into possible marketing campaign directions, and shows potential repeat-buying trends. The opinions and self-evaluations provided by the respondents, when combined with the separate fitbit user dataset could provide powerful insight into Bellabeat's target market.