BellaBeat CaseStude | R

Tejaswi

2023-03-15

# About Company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women.

## Characters and products

### \* Characters

* **Urška Sršen :** Bellabeat’s cofounder and Chief Creative Officer
* **Sando Mur :** Mathematician and Bellabeat’s cofounder; key member of the Bellabeat executive team
* **Bellabeat marketing analytics team :** A team of data analysts.

### \* Products

* **Bellabeat app**
* **Leaf :** bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress.
* **Time watch :**The Time watch connects to the Bellabeat app to provide you with insights into your daily wellness.
* **Spring :** This is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels.
* **Bellabeat membership :** Bellabeat also offers a subscription-based membership program for users. Membership gives users 24/7 access to fully personalized guidance on nutrition, activity, sleep, health and beauty, and mindfulness based on their lifestyle and goals.

### Business Task

The Data Analyst team have be asked to gain insights into any trends in the use of non-Bellabeat smart devices utilising publicly available data.

If any trends are noted, the team is to look at how to apply these trends and advise on how it could apply to Bellabeat customers in order to influence future marketing strategies.

These findings and a high-level recommendation are then to be reported to the Bellabeat Executive Team.

### Executive Summery

* Key finding - Participants are not sleeping in time.
  + Recommendation - BellaBeat watch and app should send them notification to sleep and alarm to wake up in time and motivate them to do regular exercise and walk.
  + Recommendation - Distribute advice on good sleeping routines, possibly partner with meditation apps to help (Headspace, Calm…).
* Key finding - The average participant was not hitting their recommended goals of 10,000 steps / 30 minutes of moderate activity per day.
  + Recommendation - Remind users of recommended guidelines for steps and exercise. If possible push notifications / allow setting of goals per hour.
  + Recommendation - Display how many calories were burnt during specific time or hour.
* Key finding - The dataset was very small, with a maximum of just 33 participants sampled over less than 5 weeks.
  + Recommendation - For more accurate findings, utilise a larger dataset over a longer period. Ideally from BellaBeat customers if they are the intended target.
* Key finding - Participants were most active on a Tuesday, but least active on a Sunday.
  + Recommendation - Although it’s a day fo rest, try and get users moving on a Sunday (or even remind them to rest!). Also, congratulate them when they do hit their goals.
* Key finding - sex/gender does not play a factor in findings.
  + Recommendation - As these BellaBeat Products are focused to women, including personalized women operations should be included.
* Key finding - Participants were most active at midday and between 17:00 and 19:00.
  + Recommendation - recommend short workouts for lunchtimes and how to keep active at work.

## Analysis

### Analysis Introduction

The phases of answering this question followed Google’s 6 Data Analysis steps: **Ask, Prepare, Process, Analyse, Share,** and **Act**. As such, this report will follow that structure and address any problems, concerns or considerations along the way.

## Step 1 - Ask

The Ask phase presents two Key tasks:

* Identify the business task
* Consider key stakeholders

This leads to one Deliverable:

* A clear statement of the business task

#### Key Task - Identify the business task

The Data Analyst team have be asked to gain insights into any trends in the use of non-Bellabeat smart devices utilising publicly available data.

If any trends are noted, the team is to look at how to apply these trends to one Bellabeat device and advise on how it could apply to Bellabeat customers in order to influence future marketing strategies.

#### Key Task - Consider key stakeholders

Key stakeholders include the Executive Team (Urška Sršen and Sando Mur), the Marketing Analytics Team and the Customer Base. Reporting will be back to the Executive Team.

#### Deliverable - Business task

The Data Analyst team have be asked to gain insights into any trends in the use of non-Bellabeat smart devices utilising publicly available data.

If any trends are noted, the team is to look at how to apply these trends to one Bellabeat device and advise on how it could apply to Bellabeat customers in order to influence future marketing strategies.

## Step 2 - Prepare

The Prepare phase presents three Key tasks:

* Download the data and store it appropriately
* Identify how it’s organised
* Sort and filter the data

This leads to one Deliverable:

* A description of all data sources used

### Download the data

This project is part of Google certification course ao the data was given by Google. Since the data is very large, I am using rstudio for analysis. Before importing data to rstudio console, install packages.

knitr::opts\_chunk$set(warning = FALSE)  
install.packages("tidyverse")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

install.packages("skimr")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

install.packages("reshape2")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

install.packages("janitor")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

install.packages("ggpubr")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

library("tidyverse")

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.0  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library("skimr")  
library("dplyr")  
library("lubridate")  
library("reshape2")

##   
## Attaching package: 'reshape2'  
##   
## The following object is masked from 'package:tidyr':  
##   
## smiths

library("RColorBrewer")  
library("ggpubr")  
library("janitor")

##   
## Attaching package: 'janitor'  
##   
## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

Import the data into environment.

#upload and read csv files  
#install readr package   
  
install.packages("readr")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

library("readr")  
  
#use read.csv function to read files  
  
calorie\_hour <- read.csv("hourlyCalories\_merged.csv")  
intensity\_hour <- read.csv("hourlyIntensities\_merged.csv")  
steps\_hour <- read.csv("hourlySteps\_merged.csv")  
METs\_minute <- read.csv("minuteMETsNarrow\_merged.csv")  
heartrate\_second <- read.csv("heartrate\_seconds\_merged.csv")  
sleep\_day <-read.csv("sleepDay\_merged.csv")  
weight\_day <- read.csv("weightLogInfo\_merged.csv")  
activity\_day <- read.csv("dailyActivity\_merged.csv")

### Identify how it’s organised

#use head() to check the column names made relative sense.  
  
head(calorie\_hour)

## Id ActivityHour Calories  
## 1 1503960366 4/12/2016 12:00:00 AM 81  
## 2 1503960366 4/12/2016 1:00:00 AM 61  
## 3 1503960366 4/12/2016 2:00:00 AM 59  
## 4 1503960366 4/12/2016 3:00:00 AM 47  
## 5 1503960366 4/12/2016 4:00:00 AM 48  
## 6 1503960366 4/12/2016 5:00:00 AM 48

head(intensity\_hour)

## Id ActivityHour TotalIntensity AverageIntensity  
## 1 1503960366 4/12/2016 12:00:00 AM 20 0.333333  
## 2 1503960366 4/12/2016 1:00:00 AM 8 0.133333  
## 3 1503960366 4/12/2016 2:00:00 AM 7 0.116667  
## 4 1503960366 4/12/2016 3:00:00 AM 0 0.000000  
## 5 1503960366 4/12/2016 4:00:00 AM 0 0.000000  
## 6 1503960366 4/12/2016 5:00:00 AM 0 0.000000

head(heartrate\_second)

## Id Time Value  
## 1 2022484408 4/12/2016 7:21:00 AM 97  
## 2 2022484408 4/12/2016 7:21:05 AM 102  
## 3 2022484408 4/12/2016 7:21:10 AM 105  
## 4 2022484408 4/12/2016 7:21:20 AM 103  
## 5 2022484408 4/12/2016 7:21:25 AM 101  
## 6 2022484408 4/12/2016 7:22:05 AM 95

head(activity\_day)

## Id ActivityDate TotalSteps TotalDistance TrackerDistance  
## 1 1503960366 4/12/2016 13162 8.50 8.50  
## 2 1503960366 4/13/2016 10735 6.97 6.97  
## 3 1503960366 4/14/2016 10460 6.74 6.74  
## 4 1503960366 4/15/2016 9762 6.28 6.28  
## 5 1503960366 4/16/2016 12669 8.16 8.16  
## 6 1503960366 4/17/2016 9705 6.48 6.48  
## LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance  
## 1 0 1.88 0.55  
## 2 0 1.57 0.69  
## 3 0 2.44 0.40  
## 4 0 2.14 1.26  
## 5 0 2.71 0.41  
## 6 0 3.19 0.78  
## LightActiveDistance SedentaryActiveDistance VeryActiveMinutes  
## 1 6.06 0 25  
## 2 4.71 0 21  
## 3 3.91 0 30  
## 4 2.83 0 29  
## 5 5.04 0 36  
## 6 2.51 0 38  
## FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories  
## 1 13 328 728 1985  
## 2 19 217 776 1797  
## 3 11 181 1218 1776  
## 4 34 209 726 1745  
## 5 10 221 773 1863  
## 6 20 164 539 1728

head(METs\_minute)

## Id ActivityMinute METs  
## 1 1503960366 4/12/2016 12:00:00 AM 10  
## 2 1503960366 4/12/2016 12:01:00 AM 10  
## 3 1503960366 4/12/2016 12:02:00 AM 10  
## 4 1503960366 4/12/2016 12:03:00 AM 10  
## 5 1503960366 4/12/2016 12:04:00 AM 10  
## 6 1503960366 4/12/2016 12:05:00 AM 12

head(sleep\_day)

## Id SleepDay TotalSleepRecords TotalMinutesAsleep  
## 1 1503960366 4/12/2016 12:00:00 AM 1 327  
## 2 1503960366 4/13/2016 12:00:00 AM 2 384  
## 3 1503960366 4/15/2016 12:00:00 AM 1 412  
## 4 1503960366 4/16/2016 12:00:00 AM 2 340  
## 5 1503960366 4/17/2016 12:00:00 AM 1 700  
## 6 1503960366 4/19/2016 12:00:00 AM 1 304  
## TotalTimeInBed  
## 1 346  
## 2 407  
## 3 442  
## 4 367  
## 5 712  
## 6 320

head(steps\_hour)

## Id ActivityHour StepTotal  
## 1 1503960366 4/12/2016 12:00:00 AM 373  
## 2 1503960366 4/12/2016 1:00:00 AM 160  
## 3 1503960366 4/12/2016 2:00:00 AM 151  
## 4 1503960366 4/12/2016 3:00:00 AM 0  
## 5 1503960366 4/12/2016 4:00:00 AM 0  
## 6 1503960366 4/12/2016 5:00:00 AM 0

head(weight\_day)

## Id Date WeightKg WeightPounds Fat BMI  
## 1 1503960366 5/2/2016 11:59:59 PM 52.6 115.9631 22 22.65  
## 2 1503960366 5/3/2016 11:59:59 PM 52.6 115.9631 NA 22.65  
## 3 1927972279 4/13/2016 1:08:52 AM 133.5 294.3171 NA 47.54  
## 4 2873212765 4/21/2016 11:59:59 PM 56.7 125.0021 NA 21.45  
## 5 2873212765 5/12/2016 11:59:59 PM 57.3 126.3249 NA 21.69  
## 6 4319703577 4/17/2016 11:59:59 PM 72.4 159.6147 25 27.45  
## IsManualReport LogId  
## 1 True 1.462234e+12  
## 2 True 1.462320e+12  
## 3 False 1.460510e+12  
## 4 True 1.461283e+12  
## 5 True 1.463098e+12  
## 6 True 1.460938e+12

I’m interested to see if the same amount of participants took part in each activity. As each data set contains a unique Id for each user, I can use distinct to find this out.

#check if all users took part in all activities  
n\_distinct(heartrate\_second$Id)

## [1] 14

n\_distinct(intensity\_hour$Id)

## [1] 33

n\_distinct(METs\_minute$Id)

## [1] 33

n\_distinct(sleep\_day$Id)

## [1] 24

n\_distinct(steps\_hour$Id)

## [1] 33

n\_distinct(weight\_day$Id)

## [1] 8

* Not all users participated in all activities.
* As Weight\_day has only 8 participants and heartrate\_second has only 14 participants, we can remove these two table data.
* METs\_minute has no more importance. Therefore, remove minute\_MET Table also.

remove(minute\_MET, weight\_day, heartrate\_second)

## Step 3 - Process

#### The Process phase presents four Key tasks:

Choose your tools Check the data for errors Transform the data so you can work with it effectively Document the cleaning process

#### This leads to one Deliverable:

Documentation of any cleaning or manipulation of the data

Renaming date columns to ‘date’ and split date and time into two columns

activity\_day <- activity\_day %>%  
 rename(Date = ActivityDate)  
  
sleep\_day <- sleep\_day %>%  
 rename(Date = SleepDay)  
  
activity\_day$Date=as.POSIXct(activity\_day$Date, format="%m/%d/%Y", tz=Sys.timezone())  
activity\_day$Date <- format(activity\_day$Date, format = "%m/%d/%y")  
  
calorie\_hour$ActivityHour=as.POSIXct(calorie\_hour$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
calorie\_hour$Time <- format(calorie\_hour$ActivityHour, format = "%H:%M:%S")  
calorie\_hour$Date <- format(calorie\_hour$ActivityHour, format = "%m/%d/%y")  
  
head(calorie\_hour)  
  
steps\_hour$ActivityHour=as.POSIXct(steps\_hour$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
steps\_hour$Time <- format(steps\_hour$ActivityHour, format = "%H:%M:%S")  
steps\_hour$Date <- format(steps\_hour$ActivityHour, format = "%m/%d/%y")  
  
head(steps\_hour)  
  
sleep\_day$Date=as.POSIXct(sleep\_day$Date, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
sleep\_day$Time <- format(sleep\_day$Date, format = "%H:%M:%S")  
sleep\_day$Date <- format(sleep\_day$Date, format = "%m/%d/%y")  
  
head(sleep\_day)  
  
intensity\_hour$ActivityHour=as.POSIXct(intensity\_hour$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
intensity\_hour$Time <- format(intensity\_hour$ActivityHour, format = "%H:%M:%S")  
intensity\_hour$Date <- format(intensity\_hour$ActivityHour, format = "%m/%d/%y")  
  
head(intensity\_hour)

## Step 4 - Analyse

#### The Process phase presents five Key tasks:

Aggregate your data so it’s useful and accessible Organise and format your data Perform calculations Indentify trends and relationships

#### This leads to one Deliverable:

A summary of your analysis

#Analysis  
  
#Once again check the data  
head(intensity\_hour)

## Id ActivityHour TotalIntensity AverageIntensity Time  
## 1 1503960366 2016-04-12 00:00:00 20 0.333333 00:00:00  
## 2 1503960366 2016-04-12 01:00:00 8 0.133333 01:00:00  
## 3 1503960366 2016-04-12 02:00:00 7 0.116667 02:00:00  
## 4 1503960366 2016-04-12 03:00:00 0 0.000000 03:00:00  
## 5 1503960366 2016-04-12 04:00:00 0 0.000000 04:00:00  
## 6 1503960366 2016-04-12 05:00:00 0 0.000000 05:00:00  
## Date  
## 1 04/12/16  
## 2 04/12/16  
## 3 04/12/16  
## 4 04/12/16  
## 5 04/12/16  
## 6 04/12/16

head(steps\_hour)

## Id ActivityHour StepTotal Time Date  
## 1 1503960366 2016-04-12 00:00:00 373 00:00:00 04/12/16  
## 2 1503960366 2016-04-12 01:00:00 160 01:00:00 04/12/16  
## 3 1503960366 2016-04-12 02:00:00 151 02:00:00 04/12/16  
## 4 1503960366 2016-04-12 03:00:00 0 03:00:00 04/12/16  
## 5 1503960366 2016-04-12 04:00:00 0 04:00:00 04/12/16  
## 6 1503960366 2016-04-12 05:00:00 0 05:00:00 04/12/16

head(calorie\_hour)

## Id ActivityHour Calories Time Date  
## 1 1503960366 2016-04-12 00:00:00 81 00:00:00 04/12/16  
## 2 1503960366 2016-04-12 01:00:00 61 01:00:00 04/12/16  
## 3 1503960366 2016-04-12 02:00:00 59 02:00:00 04/12/16  
## 4 1503960366 2016-04-12 03:00:00 47 03:00:00 04/12/16  
## 5 1503960366 2016-04-12 04:00:00 48 04:00:00 04/12/16  
## 6 1503960366 2016-04-12 05:00:00 48 05:00:00 04/12/16

# Aggregate, organize and format your data  
  
activity\_hour <- merge(calorie\_hour,intensity\_hour, by=c("Id","ActivityHour", "Date", "Time"))  
activity\_hour <- merge(activity\_hour, steps\_hour, by=c("Id", "ActivityHour", "Date", "Time"))  
remove(calorie\_hour,intensity\_hour,steps\_hour)  
  
head(activity\_hour)

## Id ActivityHour Date Time Calories TotalIntensity  
## 1 1503960366 2016-04-12 00:00:00 04/12/16 00:00:00 81 20  
## 2 1503960366 2016-04-12 01:00:00 04/12/16 01:00:00 61 8  
## 3 1503960366 2016-04-12 02:00:00 04/12/16 02:00:00 59 7  
## 4 1503960366 2016-04-12 03:00:00 04/12/16 03:00:00 47 0  
## 5 1503960366 2016-04-12 04:00:00 04/12/16 04:00:00 48 0  
## 6 1503960366 2016-04-12 05:00:00 04/12/16 05:00:00 48 0  
## AverageIntensity StepTotal  
## 1 0.333333 373  
## 2 0.133333 160  
## 3 0.116667 151  
## 4 0.000000 0  
## 5 0.000000 0  
## 6 0.000000 0

activity\_day %>%   
 select(TotalSteps,  
 TotalDistance,  
 Calories) %>%  
 summary()

## TotalSteps TotalDistance Calories   
## Min. : 0 Min. : 0.000 Min. : 0   
## 1st Qu.: 3790 1st Qu.: 2.620 1st Qu.:1828   
## Median : 7406 Median : 5.245 Median :2134   
## Mean : 7638 Mean : 5.490 Mean :2304   
## 3rd Qu.:10727 3rd Qu.: 7.713 3rd Qu.:2793   
## Max. :36019 Max. :28.030 Max. :4900

From this we can see that the participants were doing around 7400 steps on average, 5Km and burning 2300 calories a day.

activity\_hour %>%   
 select(StepTotal,  
 Calories) %>%  
 summary()

## StepTotal Calories   
## Min. : 0.0 Min. : 42.00   
## 1st Qu.: 0.0 1st Qu.: 63.00   
## Median : 40.0 Median : 83.00   
## Mean : 320.2 Mean : 97.39   
## 3rd Qu.: 357.0 3rd Qu.:108.00   
## Max. :10554.0 Max. :948.00

From this people who do 320 steps are burning 97 calories an hour For more accurate analysis, average must be calculated only using awake hours

sleep\_day$SleepDifference <- (sleep\_day$TotalTimeInBed - sleep\_day$TotalMinutesAsleep)  
summary(sleep\_day$SleepDifference)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 17.00 25.00 39.17 40.00 371.00

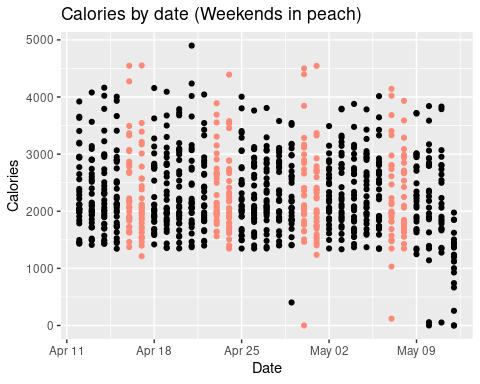
This shows that, on average, participants were in bed for almost 40 minutes more than they slept. This could point to difficulty falling asleep or staying asleep. Recommendation - Distribute advice on good sleeping routines, possibly partner with meditation apps (Headspace, Calm…).

#### Key Task - Indentify trends and relationships

activity\_day$Date <- as.Date(activity\_day$Date, "%m/%d/%y")  
activity\_day$weekday <- weekdays(activity\_day$Date)   
activity\_day$Saturday <- ifelse(activity\_day$weekday == "Saturday", "TRUE", "FALSE")  
activity\_day$Sunday <- ifelse(activity\_day$weekday == "Sunday", "TRUE", "FALSE")  
activity\_day$weekend <- ifelse(activity\_day$Saturday == TRUE | activity\_day$Sunday == TRUE, "TRUE", "FALSE")

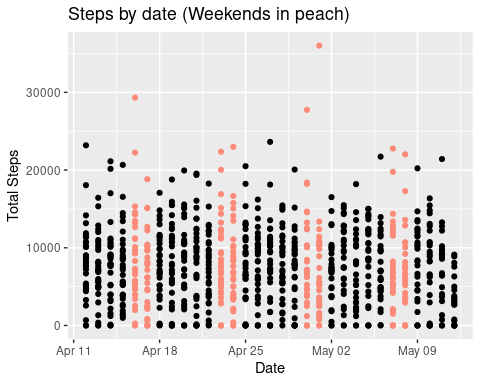
#### Calories Vs Date

ggplot(data=activity\_day, aes(x=Date, y=Calories)) + ggtitle("Calories by date (Weekends in peach)") + geom\_point(color=ifelse(activity\_day$weekend == TRUE, '#FB8A78', 'Black'))



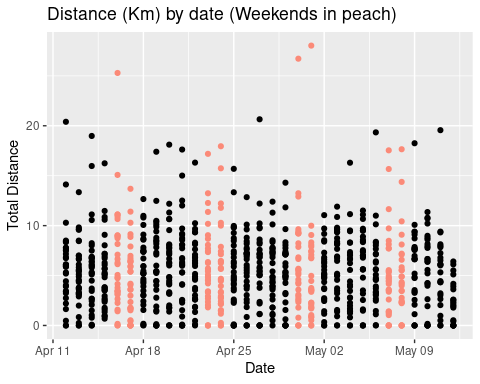
#### Steps Vs Date

ggplot(data=activity\_day, aes(x=Date, y=TotalSteps)) + ggtitle("Steps by date (Weekends in peach)") + ylab("Total Steps") + geom\_point(color=ifelse(activity\_day$weekend == TRUE, '#FB8A78', 'Black'))



#### Distance (Km) by date

ggplot(data=activity\_day, aes(x=Date, y=TotalDistance)) + ggtitle("Distance (Km) by date (Weekends in peach)") + ylab("Total Distance") + geom\_point(color=ifelse(activity\_day$weekend == TRUE, '#FB8A78', 'Black'))



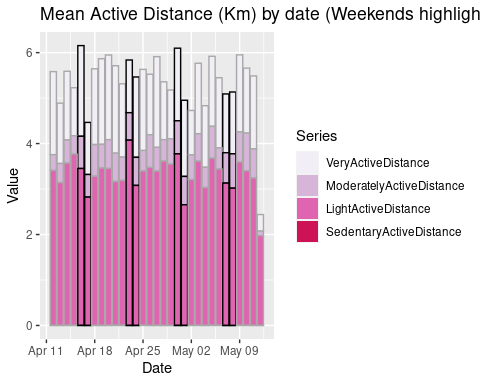
These plots do not show anything significant, with a fairly even spread of activity throughout the week. It is worth noting the 3 peach outliers with very high data points on the Steps and Distance graphs. This will skew any later averages. (Note also a drop-off in the last column throughout; possibly a cut-off in the data?)

Next we can plot the mean of Activity Distance, Activity Minutes and “Moderate” Activity onto bar graphs, again highlighting the weekends.

active\_distance <- select(activity\_day, Date, VeryActiveDistance, ModeratelyActiveDistance, LightActiveDistance, SedentaryActiveDistance)  
active\_distance <- aggregate(x = active\_distance[c("VeryActiveDistance","ModeratelyActiveDistance", "LightActiveDistance","SedentaryActiveDistance")],  
 FUN = mean,  
 by = list(Date = active\_distance$Date))  
active\_distance\_pivot <- melt(active\_distance, id.vars = "Date", variable.name = "Series")  
active\_distance\_pivot$weekday <- weekdays(active\_distance\_pivot$Date)   
active\_distance\_pivot$Saturday <- ifelse(active\_distance\_pivot$weekday == "Saturday", "TRUE", "FALSE")  
active\_distance\_pivot$Sunday <- ifelse(active\_distance\_pivot$weekday == "Sunday", "TRUE", "FALSE")  
active\_distance\_pivot$weekend <- ifelse(active\_distance\_pivot$Saturday == TRUE | active\_distance\_pivot$Sunday == TRUE, "TRUE", "FALSE")  
  
active\_minutes <- select(activity\_day, Date, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes)  
active\_minutes <- aggregate(x = active\_minutes[c("VeryActiveMinutes","FairlyActiveMinutes", "LightlyActiveMinutes","SedentaryMinutes")],FUN = mean,  
 by = list(Date = active\_minutes$Date))  
active\_minutes\_pivot <- melt(active\_minutes, id.vars = "Date", variable.name = "Series")  
active\_minutes\_pivot$weekday <- weekdays(active\_minutes\_pivot$Date)   
active\_minutes\_pivot$Saturday <- ifelse(active\_minutes\_pivot$weekday == "Saturday", "TRUE", "FALSE")  
active\_minutes\_pivot$Sunday <- ifelse(active\_minutes\_pivot$weekday == "Sunday", "TRUE", "FALSE")  
active\_minutes\_pivot$weekend <- ifelse(active\_minutes\_pivot$Saturday == TRUE | active\_minutes\_pivot$Sunday == TRUE, "TRUE", "FALSE")  
  
active\_minutes\_moderate <- aggregate(x = active\_minutes[c("VeryActiveMinutes","FairlyActiveMinutes")],  
 FUN = mean,  
 by = list(Date = active\_minutes$Date))  
active\_minutes\_moderate\_pivot <- melt(active\_minutes\_moderate, id.vars = "Date", variable.name = "Series")  
active\_minutes\_moderate\_pivot$weekday <- weekdays(active\_minutes\_moderate\_pivot$Date)   
active\_minutes\_moderate\_pivot$Saturday <- ifelse(active\_minutes\_moderate\_pivot$weekday == "Saturday", "TRUE", "FALSE")  
active\_minutes\_moderate\_pivot$Sunday <- ifelse(active\_minutes\_moderate\_pivot$weekday == "Sunday", "TRUE", "FALSE")  
active\_minutes\_moderate\_pivot$weekend <- ifelse(active\_minutes\_moderate\_pivot$Saturday == TRUE | active\_minutes\_moderate\_pivot$Sunday == TRUE, "TRUE", "FALSE")  
  
active\_minutes\_nonsedentary <- select(activity\_day, Date, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes)  
active\_minutes\_nonsedentary <- aggregate(x = active\_minutes\_nonsedentary[c("VeryActiveMinutes","FairlyActiveMinutes", "LightlyActiveMinutes")],  
 FUN = mean,  
 by = list(Date = active\_minutes\_nonsedentary$Date))  
active\_minutes\_nonsedentary\_pivot <- melt(active\_minutes\_nonsedentary, id.vars = "Date", variable.name = "Series")  
active\_minutes\_nonsedentary\_pivot$weekday <- weekdays(active\_minutes\_nonsedentary\_pivot$Date)   
active\_minutes\_nonsedentary\_pivot$Saturday <- ifelse(active\_minutes\_nonsedentary\_pivot$weekday == "Saturday", "TRUE", "FALSE")  
active\_minutes\_nonsedentary\_pivot$Sunday <- ifelse(active\_minutes\_nonsedentary\_pivot$weekday == "Sunday", "TRUE", "FALSE")  
active\_minutes\_nonsedentary\_pivot$weekend <- ifelse(active\_minutes\_nonsedentary\_pivot$Saturday == TRUE | active\_minutes\_nonsedentary\_pivot$Sunday == TRUE, "TRUE", "FALSE")  
active\_minutes\_nonsedentary\_pivot$order = factor(active\_minutes\_nonsedentary\_pivot$weekday, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

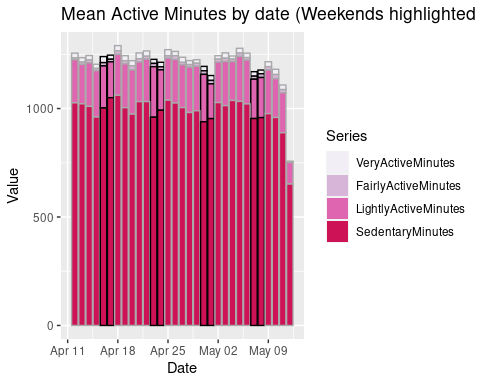
#### Mean Active Distance (Km) by date

ggplot(active\_distance\_pivot, aes(fill=Series, y=value, x=Date, color=weekend)) + geom\_bar(position="stack", stat="identity") + ggtitle("Mean Active Distance (Km) by date (Weekends highlighted black)") + ylab("Value") + scale\_fill\_brewer(palette = "PuRd") + scale\_colour\_manual(values=c("FALSE"="Dark Grey","TRUE"="Black"),guide="none")



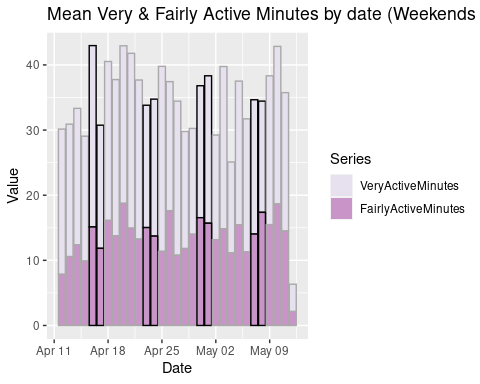
#### Mean Active Minutes by date

ggplot(active\_minutes\_pivot, aes(fill=Series, y=value, x=Date, color=weekend)) + geom\_bar(position="stack", stat="identity") + ggtitle("Mean Active Minutes by date (Weekends highlighted black)") + ylab("Value") + scale\_fill\_brewer(palette = "PuRd") + scale\_colour\_manual(values=c("FALSE"="Dark Grey","TRUE"="Black"),guide="none")



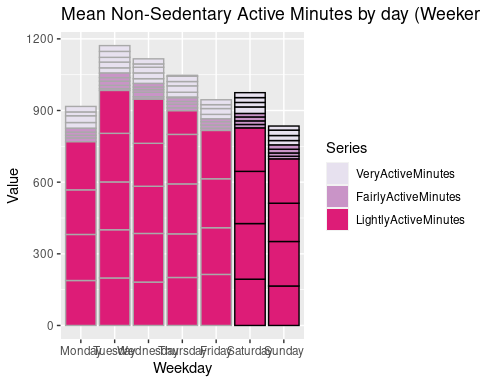
#### Mean Very & Fairly Active Minutes by date

ggplot(active\_minutes\_moderate\_pivot, aes(fill=Series, y=value, x=Date, color=weekend)) + geom\_bar(position="stack", stat="identity") + ggtitle("Mean Very & Fairly Active Minutes by date (Weekends highlighted black)") + ylab("Value") + scale\_fill\_brewer(palette = "PuRd") + scale\_colour\_manual(values=c("FALSE"="Dark Grey","TRUE"="Black"),guide="none")



####Mean Non-Sedentary Active Minutes by day

ggplot(active\_minutes\_nonsedentary\_pivot, aes(fill=Series, y=value, x=order, color=weekend)) + geom\_bar(position="stack", stat="identity") + ggtitle("Mean Non-Sedentary Active Minutes by day (Weekends highlighted black)") + ylab("Value") + xlab("Weekday") + scale\_fill\_brewer(palette = "PuRd") + scale\_colour\_manual(values=c("FALSE"="Dark Grey","TRUE"="Black"),guide="none")



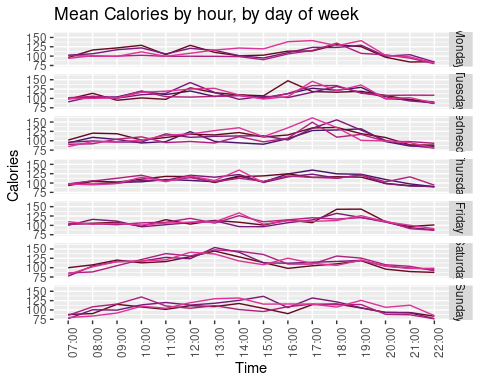
Moving on to activity\_hour dataset, looking at similar plots. As participants don’t exercise whilst they sleep I’m going to remove hours they’re likely to be asleep (2300hrs to 0600hrs). Again, it would be good to be able to highlight weekends:

activity\_hour\_waking <- activity\_hour[!(activity\_hour$Time=="23:00:00" | activity\_hour$Time=="00:00:00" | activity\_hour$Time=="01:00:00"| activity\_hour$Time=="02:00:00"| activity\_hour$Time=="03:00:00" | activity\_hour$Time=="04:00:00"| activity\_hour$Time=="05:00:00"| activity\_hour$Time=="06:00:00"),]

activity\_hour\_waking\_mean <- aggregate(x = activity\_hour\_waking[c("Calories", "TotalIntensity", "AverageIntensity", "StepTotal")],  
 FUN = mean,  
 by = list(ActivityHour = activity\_hour\_waking$ActivityHour))  
  
activity\_hour\_waking\_mean$weekday <- weekdays(activity\_hour\_waking\_mean$ActivityHour)   
activity\_hour\_waking\_mean$Saturday <- ifelse(activity\_hour\_waking\_mean$weekday == "Saturday", "TRUE", "FALSE")  
activity\_hour\_waking\_mean$Sunday <- ifelse(activity\_hour\_waking\_mean$weekday == "Sunday", "TRUE", "FALSE")  
activity\_hour\_waking\_mean$weekend <- ifelse(activity\_hour\_waking\_mean$Saturday == TRUE | activity\_hour\_waking\_mean$Sunday == TRUE, "TRUE", "FALSE")  
activity\_hour\_waking\_mean <- separate(data = activity\_hour\_waking\_mean, col = ActivityHour, into = c("Date", "Time"), sep = " ", remove = F)  
activity\_hour\_waking\_mean$Time <- substr(activity\_hour\_waking\_mean$Time,1,nchar(activity\_hour\_waking\_mean$Time)-3)  
  
activity\_hour\_waking\_mean$week <- lubridate::week(activity\_hour\_waking\_mean$ActivityHour)  
activity\_hour\_waking\_mean$facet = factor(activity\_hour\_waking\_mean$weekday, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))  
  
activity\_hour\_waking\_mean\_week15 <- filter(activity\_hour\_waking\_mean, week == 15)  
activity\_hour\_waking\_mean\_week16 <- filter(activity\_hour\_waking\_mean, week == 16)  
activity\_hour\_waking\_mean\_week17 <- filter(activity\_hour\_waking\_mean, week == 17)  
activity\_hour\_waking\_mean\_week18 <- filter(activity\_hour\_waking\_mean, week == 18)  
activity\_hour\_waking\_mean\_week19 <- filter(activity\_hour\_waking\_mean, week == 19)

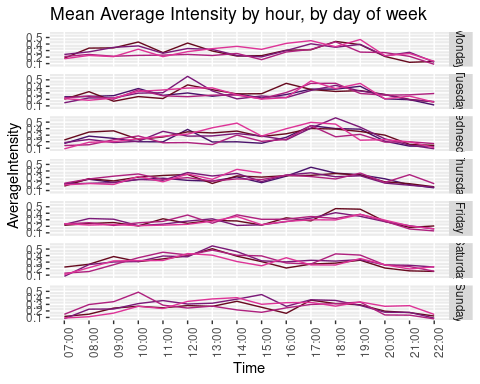
#### Mean Calories by hour, by day of week

ggplot() +   
 geom\_line(data = activity\_hour\_waking\_mean\_week15, aes(x = Time, y = Calories, group = 1), colour="#49126A") + facet\_grid("facet", scales="free\_x", space="free\_y") +   
 geom\_line(data = activity\_hour\_waking\_mean\_week16, aes(x = Time, y = Calories, group = 1), colour="#670A1F") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week17, aes(x = Time, y = Calories, group = 1), colour="#7A1977") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week18, aes(x = Time, y = Calories, group = 1), colour="#AE217E") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week19, aes(x = Time, y = Calories, group = 1), colour="#DD3497") + facet\_grid("facet", scales="free\_x", space="free\_y") + theme(axis.text.x = element\_text(angle = 90))+ ggtitle("Mean Calories by hour, by day of week")



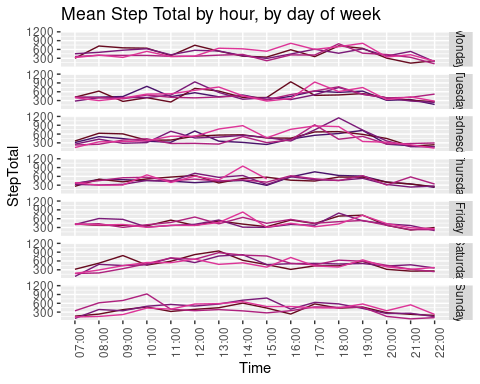
#### Mean Average Intensity by hour, by day of week

ggplot() +   
 geom\_line(data = activity\_hour\_waking\_mean\_week15, aes(x = Time, y = AverageIntensity, group = 1,), colour="#49126A") + facet\_grid("facet", scales="free\_x", space="free\_y") +   
 geom\_line(data = activity\_hour\_waking\_mean\_week16, aes(x = Time, y = AverageIntensity, group = 1), colour="#670A1F") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week17, aes(x = Time, y = AverageIntensity, group = 1), colour="#7A1977") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week18, aes(x = Time, y = AverageIntensity, group = 1), colour="#AE217E") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week19, aes(x = Time, y = AverageIntensity, group = 1), colour="#DD3497") + facet\_grid("facet", scales="free\_x", space="free\_y") + theme(axis.text.x = element\_text(angle = 90))+ ggtitle("Mean Average Intensity by hour, by day of week")



#### Mean Steps Total by hour, ny day of week

ggplot() +   
 geom\_line(data = activity\_hour\_waking\_mean\_week15, aes(x = Time, y = StepTotal, group = 1), colour="#49126A") + facet\_grid("facet", scales="free\_x", space="free\_y") +   
 geom\_line(data = activity\_hour\_waking\_mean\_week16, aes(x = Time, y = StepTotal, group = 1), colour="#670A1F") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week17, aes(x = Time, y = StepTotal, group = 1), colour="#7A1977") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week18, aes(x = Time, y = StepTotal, group = 1), colour="#AE217E") + facet\_grid("facet", scales="free\_x", space="free\_y") +  
 geom\_line(data = activity\_hour\_waking\_mean\_week19, aes(x = Time, y = StepTotal, group = 1), colour="#DD3497") + facet\_grid("facet", scales="free\_x", space="free\_y") + theme(axis.text.x = element\_text(angle = 90))+ ggtitle("Mean Step Total by hour, by day of week")

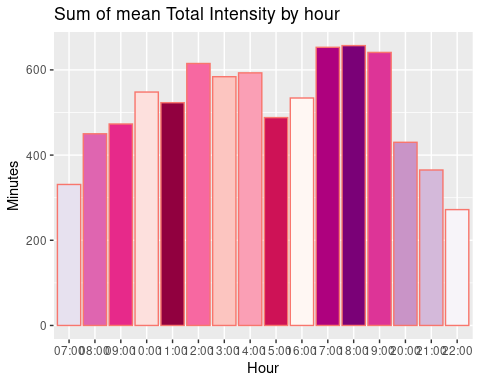


#### The most / least active waking hours of those weekdays

activity\_hour\_waking\_weekday <- activity\_hour\_waking\_mean[!(activity\_hour\_waking\_mean$weekend==TRUE),]  
activity\_hour\_waking\_weekday <- aggregate(x = activity\_hour\_waking\_mean[c("TotalIntensity")],  
 FUN = sum,  
 by = list(Time = activity\_hour\_waking\_mean$Time))  
activity\_hour\_waking\_weekday$TotalIntensity <- round(activity\_hour\_waking\_weekday$TotalIntensity)

#### Sum of mean TOtal Intensity by hour

mycolors = c(brewer.pal(name="PuRd", n = 8), brewer.pal(name="RdPu", n = 8))  
ggplot(activity\_hour\_waking\_weekday, aes(fill=as.factor(TotalIntensity), x=Time, y=TotalIntensity, color="DarkGrey")) + geom\_bar(stat="identity") + ggtitle("Sum of mean Total Intensity by hour") + ylab("Minutes") + xlab("Hour") + theme(legend.position = "none") + scale\_fill\_manual(values = mycolors)



Here we can see that the most active minutes are the 17:00 to 19:00 portion and 12:00. This points to participants working out at lunchtime and after work. Recommendation - recommend short workouts for lunchtimes and how to keep active at work.

## Step 5 - Share

The Share phase presents four Key tasks:

* Determine the best way to share your findings
* Create effective data visualization
* Present your finding
* Ensure your work is accessible

This leads to one Deliverable:

* Supporting visualizations and key findings

#### Key Task - Determine the best way to share your findings

Ordinarily the top levels of this report would be removed and placed into a Powerpoint presentation or document. As I will not be briefing, I will not carry this out. I can share this .rmd via my Kaggle profile instead.

#### Key Tasks - Create effective data visualization / Present your finding / Ensure your work is accessible

These tasks have been carried out during the creation and preparation of this report.

#### Deliverable - Supporting visualizations and key findings

Supporting visualizations and key findings can be found throughout this report and in the Executive Summary at the head of the document.

## Step 6 - Act

The Act phase presents three Key tasks:

* Create your portfolio
* Add your case study
* Practice presenting your case study to a friend or family member

This leads to one Deliverable:

* Your top high-level insights based on your analysis

#### Key Task - Practice presenting your case study to a friend or family member

Prepare to get briefed people!

#### Deliverable - Your top high-level insights based on your analysis

High-level insights can be found in the Executive Summary of this report

## Thanks and wrap-up

Thank you very much if you’ve got this far! This was my first foray into R, so I’m sure there’s plenty of errors and things I could have done easier / better. If you have any tips, please let me know as I’m dying to learn more.

An additional thanks to the Kaggle and Google for giving me this task.

Good luck to all my fellow Google Scholarship Data Analysts past, present and future.

I hope this project will open me many of job opportunities.