Análise Bellabeat - Google Capstone Project

Yuri Faria Frota

31/10/2021

# Bellabeat

Bellabeat é uma empresa de tecnologia responsável por produzir aparelhos smart especificamente para mulheres.

O background artístico de sua fundadora, Urška Sršen, proporcionou a Bellabeat a capacidade de criar belos designs para seus aparelhos, capazes de informar e inspirar mulheres de todo o mundo. Coletando dados referentes à características como, sono, atividades, estresse, e fertilidade, permitindo o empoderamento das mulheres através do conhecimento à respeito de sua saúde e hábitos diários.

Fundada em 2013, Bellabeat cresceu rapidamente e se consolidou como uma empresa de saúde da mulher baseada em tecnologias direcionadas.

Sršen sabe que uma análise dos dados pode revelar importantes insights e oportunidades de crescimento. Foi demandado ao time de análise de marketing uma análise dos dados de outros aparelhos smart (similares aos produtos da Bellabeat), e através dos insights obtidos, buscar saber mais a respeito de como as pessoas estão utilizando seus aparelhos, e como podemos direcionar essas informações para algum produto da Bellabeat.

Com as informações obtidas, ela gostaria de boas indicações de como essas tendências podem ser informativas para o departamento de marketing e estratégia.

Acesse o site para mais informações <https://bellabeat.com/>

## FASE 1 - Pergunta (Ask)

Através dos dados de equipamentos smart (como gasto calórico, tempo de atividades, dados diários referentes à qualidade do sono, entre outros), será possivel obter insights sobre como melhorar os equipamentos da Bellabeat e atender melhor as demandas de seus clientes.

Desta forma, os clientes Bellabeat terão mais conhecimento sobre seus hábitos, seu corpo, e sobre como promover uma vida saudável, proporcionando melhor qualidade de vida.

A análise desses dados pode ajudar a Bellabeat a se destacar no mercado, sendo capaz de oferecer demandas antes não ofertadas.

Teremos como objetivos principais desta análise, fornecer:

* Novos insights através de análise de dados de outros equipamentos
* Padrões e tendencias provindas dos dados
* Propostas de melhorias para os equipamentos Bellabeat
* Possibilidades de novas demandas, a partir da análise dos dados
* Informações valiosas para o departamento de estratégia e marketing

Utilizaremos a joia **Leaf**, do catálogo de produtos da Bellabeat, para nos basearmos como referência neste estudo. <https://bellabeat.com/catalog/>

## FASE 2 - Preparação (Prepare)

Esta fase consiste no início do processo de análise. Checaremos quais são os dados disponíveis, sua credibilidade, seus formatos, e quais relações podemos obter entre eles. Para saber à respeito da qualidade do banco de dados em questão, utilizaremos a metodologia ROCCC, na qual visa cinco critérios básicos para garantir a credibilidade dos dados, onde o set de dados deve ser:

#### Confiável (Reliability):

Devido a pequena amostragem (30 participantes), e a falta de outras variaveis estatísticas, restringem a possibilidade de análises mais aprofundadas.

#### Original (Originality):

Estes dados nao foram coletados originalmente pelo fabricante, e sim pela Amazon Mechanical Murk.

#### Abrangente (Comprehensiveness):

Os dados não informam sexo, idade, e outras condições que poderiam gerar resultados mais precisos. Para esta análise, Não levar em conta essas características pode tornar o resutlado enviesado.

#### Atual (Current):

Os dados foram coletados em 2016, e podem gerar informações desatualizadas.

#### Citado (Cited):

Dados coletados pela Amazon Mechanical Murk.

Após as observações acima, podemos considerar uma escassez de informações para análises mais precisas. Desta forma, iremos obter alguns insights referentes a esta análise, e iremos tomar os resultados obtidos como um ponto inicial para próximas análises.

## Configurando o ambiente

instalando e carregando os pacotes que serao utilizados.

library(readr)  
library(tidyverse)

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

## v ggplot2 3.3.5 v dplyr 1.0.7  
## v tibble 3.1.5 v stringr 1.4.0  
## v tidyr 1.1.4 v forcats 0.5.1  
## v purrr 0.3.4

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(sqldf)

## Carregando pacotes exigidos: gsubfn

## Carregando pacotes exigidos: proto

## Carregando pacotes exigidos: RSQLite

library(ggplot2)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

## Importando os dados

sleepDay\_merged <-read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/sleepDay\_merged.csv")  
weightLogInfo\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/weightLogInfo\_merged.csv")  
minuteSleep\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/minuteSleep\_merged.csv")  
minuteStepsNarrow\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/minuteStepsNarrow\_merged.csv")  
minuteIntensitiesWide\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/minuteIntensitiesWide\_merged.csv")  
minuteMETsNarrow\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/minuteMETsNarrow\_merged.csv")  
minuteCaloriesWide\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/minuteCaloriesWide\_merged.csv")  
minuteIntensitiesNarrow\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/minuteIntensitiesNarrow\_merged.csv")  
hourlyCalories\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/hourlyCalories\_merged.csv")  
hourlyIntensities\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/hourlyIntensities\_merged.csv")  
hourlySteps\_merged <-read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/hourlySteps\_merged.csv")  
minuteCaloriesNarrow\_merged <- read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/minuteCaloriesNarrow\_merged.csv")  
dailyactivity\_merged <-read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/dailyActivity\_merged.csv")  
dailyCalories\_merged <-read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/dailyCalories\_merged.csv")  
dailyIntensities\_merged <-read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/dailyIntensities\_merged.csv")  
dailySteps\_merged <-read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/dailySteps\_merged.csv")  
heartrate\_seconds\_merged <-read\_csv("C:/Users/Yuri/Documents/capstone project/fitbit\_fitnesstrack\_data/heartrate\_seconds\_merged.csv")

## Checando os dados

Tomaremos como exemplo o set de atividade diária (‘dailyactivity\_merged’) para demonstrar as inspeções realizadas.

colnames(dailyactivity\_merged)

## [1] "Id" "ActivityDate"   
## [3] "TotalSteps" "TotalDistance"   
## [5] "TrackerDistance" "LoggedActivitiesDistance"  
## [7] "VeryActiveDistance" "ModeratelyActiveDistance"  
## [9] "LightActiveDistance" "SedentaryActiveDistance"   
## [11] "VeryActiveMinutes" "FairlyActiveMinutes"   
## [13] "LightlyActiveMinutes" "SedentaryMinutes"   
## [15] "Calories"

head(dailyactivity\_merged)

## # A tibble: 6 x 15  
## Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitie~  
## <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 1.50e9 4/12/2016 13162 8.5 8.5 0  
## 2 1.50e9 4/13/2016 10735 6.97 6.97 0  
## 3 1.50e9 4/14/2016 10460 6.74 6.74 0  
## 4 1.50e9 4/15/2016 9762 6.28 6.28 0  
## 5 1.50e9 4/16/2016 12669 8.16 8.16 0  
## 6 1.50e9 4/17/2016 9705 6.48 6.48 0  
## # ... with 9 more variables: VeryActiveDistance <dbl>,  
## # ModeratelyActiveDistance <dbl>, LightActiveDistance <dbl>,  
## # SedentaryActiveDistance <dbl>, VeryActiveMinutes <dbl>,  
## # FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,  
## # SedentaryMinutes <dbl>, Calories <dbl>

nrow(dailyactivity\_merged)

## [1] 940

glimpse(dailyactivity\_merged)

## Rows: 940  
## Columns: 15  
## $ Id <dbl> 1503960366, 1503960366, 1503960366, 150396036~  
## $ ActivityDate <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~  
## $ TotalSteps <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019~  
## $ TotalDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~  
## $ TrackerDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~  
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~  
## $ VeryActiveDistance <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~  
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~  
## $ LightActiveDistance <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~  
## $ SedentaryActiveDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~  
## $ VeryActiveMinutes <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~  
## $ FairlyActiveMinutes <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~  
## $ LightlyActiveMinutes <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~  
## $ SedentaryMinutes <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~  
## $ Calories <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~

**Utilizando as mesmas funções descritas acima para todos sets de dados abaixo.**

*Dias de sono (sleepDay\_merged)*

*Intensidade diária (dailyIntensities\_merged)*

*Calorias diárias (dailyCalories\_merged)*

*Registro de peso (weightLogInfo\_merged)*

*Passos diários (dailySteps\_merged)*

## Verificando a integridade dos dados

**Vamos verificar quantas pessoas analisadas estão presentes em casa set de dados, separando apenas a quantidade de Ids distintos, e contando as linhas resultantes.**

teste1 <- sqldf ('SELECT DISTINCT Id FROM "dailyCalories\_merged"')  
nrow(teste1)

## [1] 33

teste2 <- sqldf ('SELECT DISTINCT Id FROM "dailyactivity\_merged"')  
nrow(teste2)

## [1] 33

teste3 <- sqldf ('SELECT DISTINCT Id FROM "dailyIntensities\_merged"')  
nrow(teste3)

## [1] 33

teste4 <- sqldf ('SELECT DISTINCT Id FROM "dailySteps\_merged"')  
nrow(teste4)

## [1] 33

teste5 <- sqldf ('SELECT DISTINCT Id FROM "sleepDay\_merged"')  
nrow(teste5)

## [1] 24

teste6 <- sqldf ('SELECT DISTINCT Id FROM "weightLogInfo\_merged"')  
nrow(teste6)

## [1] 8

**Verificando a quantidade de dias registrados.**

n\_distinct(dailyactivity\_merged$ActivityDate)

## [1] 31

**Durante a checagem, observou-se que ambos os sets abaixo possuem uma quantidade de registros inferior aos registros dos outros sets analisados.**

*Registro de peso (weightLogInfo\_merged)*

nrow(weightLogInfo\_merged)

## [1] 67

n\_distinct(weightLogInfo\_merged$Id)

## [1] 8

*Passos diarios (dailySteps\_merged)*

nrow(sleepDay\_merged)

## [1] 413

n\_distinct(sleepDay\_merged$Id)

## [1] 24

**Ambos sets acima se apresentam incompletos, onde nao constam todos os participantes, e com menos registros que os demais, portanto nao vamos considerá-los em nossa análise.**

**Vamos proceder com a utilização das tabelas com dados diários, e verificar se as colunas iguais possuem os mesmos dados entre as tabelas.** **Para isso utilizamos as seguintes funções:**

**‘dailyActivity\_merged’ ∩ ‘dailyCalories\_merged’**

daily\_activity1 <- dailyactivity\_merged %>% select(Id, ActivityDate, Calories)  
check1 <- sqldf ('SELECT \* FROM daily\_activity1 INTERSECT SELECT \* FROM dailyCalories\_merged')  
nrow(check1)

## [1] 940

n\_distinct(check1$Id)

## [1] 33

**‘dailyActivity\_merged’ ∩ ‘dailyIntensities\_merged’**

daily\_activity2 <- dailyactivity\_merged %>% select(Id, ActivityDate, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, VeryActiveMinutes, SedentaryActiveDistance, LightActiveDistance, ModeratelyActiveDistance, VeryActiveDistance)  
check2 <- sqldf('SELECT \* FROM daily\_activity2 INTERSECT SELECT \* FROM dailyIntensities\_merged')  
nrow(check2)

## [1] 940

n\_distinct(check2$Id)

## [1] 33

**‘dailyActivity\_merged’ ∩ ‘dailySteps\_merged’**

daily\_activity3 <- dailyactivity\_merged %>% select(Id, ActivityDate, TotalSteps)  
check3 <- sqldf('SELECT \* FROM daily\_activity3 INTERSECT SELECT \* FROM dailySteps\_merged')  
nrow(check3)

## [1] 940

n\_distinct(check3$Id)

## [1] 33

**as 940 linhas encontradas em suas intersecções (check1, check2, e check3) afirmam que as colunas entre as tabelas possuem os mesmos dados, portanto podemos nos basear na tabela ‘dailyActivity\_merged’ como um resumo geral de todos os dados diários coletados.**

## Fase 3 e 4: Processamento e análise dos dados (Process and Analyze)

**Vamos agora gerar um resumo das caracteristicas gerais dos hábitos diários para iniciar nossa análise.**

dailyactivity\_merged %>% summary()

## Id ActivityDate TotalSteps TotalDistance   
## Min. :1.504e+09 Length:940 Min. : 0 Min. : 0.000   
## 1st Qu.:2.320e+09 Class :character 1st Qu.: 3790 1st Qu.: 2.620   
## Median :4.445e+09 Mode :character Median : 7406 Median : 5.245   
## Mean :4.855e+09 Mean : 7638 Mean : 5.490   
## 3rd Qu.:6.962e+09 3rd Qu.:10727 3rd Qu.: 7.713   
## Max. :8.878e+09 Max. :36019 Max. :28.030   
## TrackerDistance LoggedActivitiesDistance VeryActiveDistance  
## Min. : 0.000 Min. :0.0000 Min. : 0.000   
## 1st Qu.: 2.620 1st Qu.:0.0000 1st Qu.: 0.000   
## Median : 5.245 Median :0.0000 Median : 0.210   
## Mean : 5.475 Mean :0.1082 Mean : 1.503   
## 3rd Qu.: 7.710 3rd Qu.:0.0000 3rd Qu.: 2.053   
## Max. :28.030 Max. :4.9421 Max. :21.920   
## ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance  
## Min. :0.0000 Min. : 0.000 Min. :0.000000   
## 1st Qu.:0.0000 1st Qu.: 1.945 1st Qu.:0.000000   
## Median :0.2400 Median : 3.365 Median :0.000000   
## Mean :0.5675 Mean : 3.341 Mean :0.001606   
## 3rd Qu.:0.8000 3rd Qu.: 4.782 3rd Qu.:0.000000   
## Max. :6.4800 Max. :10.710 Max. :0.110000   
## VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes  
## Min. : 0.00 Min. : 0.00 Min. : 0.0 Min. : 0.0   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.:127.0 1st Qu.: 729.8   
## Median : 4.00 Median : 6.00 Median :199.0 Median :1057.5   
## Mean : 21.16 Mean : 13.56 Mean :192.8 Mean : 991.2   
## 3rd Qu.: 32.00 3rd Qu.: 19.00 3rd Qu.:264.0 3rd Qu.:1229.5   
## Max. :210.00 Max. :143.00 Max. :518.0 Max. :1440.0   
## Calories   
## Min. : 0   
## 1st Qu.:1828   
## Median :2134   
## Mean :2304   
## 3rd Qu.:2793   
## Max. :4900

sleepDay\_merged %>% summary ()

## Id SleepDay TotalSleepRecords TotalMinutesAsleep  
## Min. :1.504e+09 Length:413 Min. :1.000 Min. : 58.0   
## 1st Qu.:3.977e+09 Class :character 1st Qu.:1.000 1st Qu.:361.0   
## Median :4.703e+09 Mode :character Median :1.000 Median :433.0   
## Mean :5.001e+09 Mean :1.119 Mean :419.5   
## 3rd Qu.:6.962e+09 3rd Qu.:1.000 3rd Qu.:490.0   
## Max. :8.792e+09 Max. :3.000 Max. :796.0   
## TotalTimeInBed   
## Min. : 61.0   
## 1st Qu.:403.0   
## Median :463.0   
## Mean :458.6   
## 3rd Qu.:526.0   
## Max. :961.0

**Através do resumo, podemos obter a média diária de alguns dados:**

* Passos dados : 7638
* Distância percorrida : 5.490m / 5.49 km
* Perda calórica :2304 kcal
* Minutos Sedentários : 991.2 / 16.52 horas
* Minutos ativos : 227.52 / 3.8 horas
* Minutos de sono : 419.5 / 7 horas

**Nesta breve análise já podemos deduzir algumas características dos participantes:**

* Em média, são sedentários (mais de 16 horas por dia com atividades sedentárias).
* Dormem pouco (menos do que as 8 horas habituais).

**Vamos observar alguns gráficos para obter alguns insights significativos.**

daily\_activity4 <- sqldf ('SELECT Id, ActivityDate,Calories, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDistance, (VeryActiveDistance+ModeratelyActiveDistance+LightActiveDistance) AS ActiveDistanceTotal,SedentaryActiveDistance, (VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes) AS ActiveMinutesTotal, SedentaryMinutes FROM "dailyactivity\_merged"')  
  
colnames(daily\_activity4)

## [1] "Id" "ActivityDate"   
## [3] "Calories" "TotalSteps"   
## [5] "TotalDistance" "TrackerDistance"   
## [7] "LoggedActivitiesDistance" "ActiveDistanceTotal"   
## [9] "SedentaryActiveDistance" "ActiveMinutesTotal"   
## [11] "SedentaryMinutes"

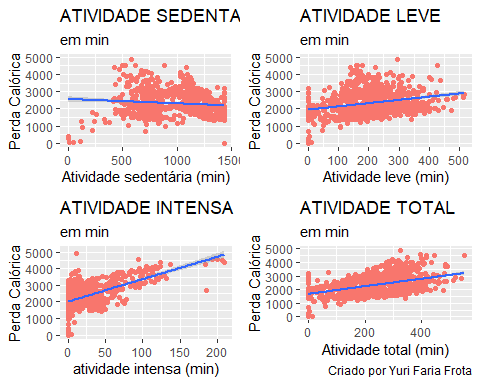
nrow(daily\_activity4)

## [1] 940

distanciamaximaplot<- ggplot(data=dailyactivity\_merged) +  
 geom\_point(mapping=aes(x=TotalDistance, y= Calories, color = 'Orange'))+  
 labs (title = "DISTÂNCIA TOTAL", subtitle = "em km ",   
 x = "Distância Total (km)", y = "Perda Calórica")+   
 theme(legend.position = "none") +  
 geom\_smooth(method=lm,mapping = aes (x=TotalDistance, y= Calories ))  
  
  
tempoleveplot <- ggplot(data=dailyactivity\_merged) +  
 geom\_point(mapping=aes(x=LightlyActiveMinutes, y= Calories, color = 'Orange'))+  
 labs (title = "ATIVIDADE LEVE", subtitle = "em min ", x = "Atividade leve (min)", y = "Perda Calórica") +  
 theme(legend.position = "none") +  
 geom\_smooth(method=lm, mapping = aes (x=LightlyActiveMinutes, y= Calories ))  
  
  
tempointensoplot <-ggplot(data=dailyactivity\_merged) +  
 geom\_point(mapping=aes(x=VeryActiveMinutes, y= Calories, color = 'Orange'))+  
 labs (title = "ATIVIDADE INTENSA", subtitle = "em min ", x = "atividade intensa (min)", y = "Perda Calórica") +   
 theme(legend.position = "none") +  
 geom\_smooth(method=lm,mapping = aes (x=VeryActiveMinutes, y= Calories ))  
  
  
tempoesedentarioplot<-ggplot(data=dailyactivity\_merged) +  
 geom\_point(mapping=aes(x=SedentaryMinutes, y= Calories, color = 'Orange'))+  
 labs (title = "ATIVIDADE SEDENTARIA", subtitle = "em min ", x = "Atividade sedentária (min)", y = "Perda Calórica") +  
 theme(legend.position = "none") +  
 geom\_smooth(method=lm, mapping = aes (x=SedentaryMinutes, y= Calories ))  
  
tempototalplot<-ggplot(data=daily\_activity4) +  
 geom\_point(mapping=aes(x=ActiveMinutesTotal, y= Calories, color = 'Orange'))+  
 labs (title = "ATIVIDADE TOTAL", subtitle = "em min ",   
 caption= "Criado por Yuri Faria Frota", x = "Atividade total (min)", y = "Perda Calórica") +  
 theme(legend.position = "none") +  
 geom\_smooth(method=lm, mapping = aes (x=ActiveMinutesTotal, y= Calories ))

grid.arrange(tempoesedentarioplot, tempoleveplot, tempointensoplot, tempototalplot, nrow = 2)

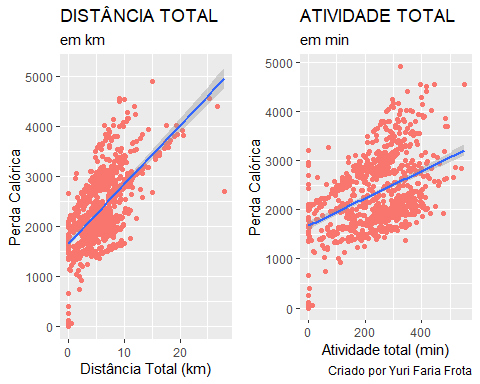
## `geom\_smooth()` using formula 'y ~ x'  
## `geom\_smooth()` using formula 'y ~ x'  
## `geom\_smooth()` using formula 'y ~ x'  
## `geom\_smooth()` using formula 'y ~ x'



**Note a diferença da perda calórica de acordo com o tipo de atividade. É evidente a importancia de manter atividades, leves, moderadas, ou intensas, para e obter um consumo calórico eficiente.**

grid.arrange(distanciamaximaplot, tempototalplot, nrow = 1)

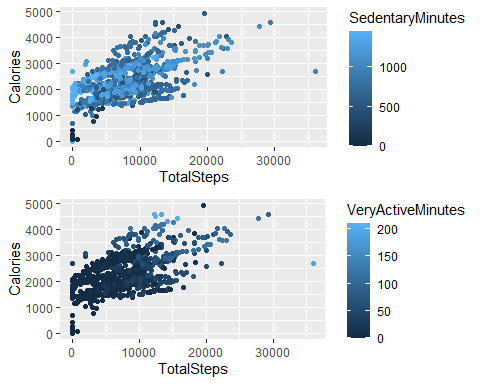
## `geom\_smooth()` using formula 'y ~ x'  
## `geom\_smooth()` using formula 'y ~ x'



**Longas caminhadas, sob atividades leves ou moderadas, também são eficientes para perder calorias.**

sedentario <- ggplot(data=dailyactivity\_merged) + geom\_point(mapping = aes (x=TotalSteps, y=Calories, color = SedentaryMinutes))  
  
leve <- ggplot(data=dailyactivity\_merged) + geom\_point(mapping = aes (x=TotalSteps, y=Calories, color = LightlyActiveMinutes))  
  
moderado <- ggplot(data=dailyactivity\_merged) + geom\_point(mapping = aes (x=TotalSteps, y=Calories, color = FairlyActiveMinutes))  
  
intenso <- ggplot(data=dailyactivity\_merged) + geom\_point(mapping = aes (x=TotalSteps, y=Calories, color = VeryActiveMinutes))

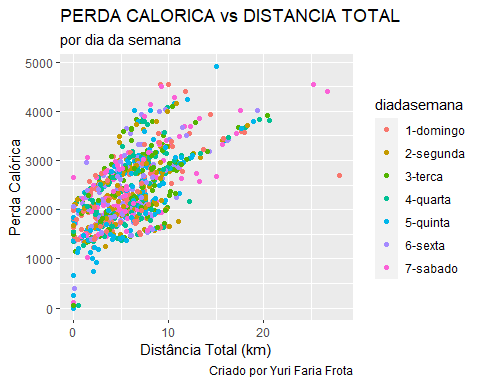
grid.arrange(sedentario, intenso, nrow = 2)



**O maior tempo de atividades dos usuários é de forma sedentária. Perceba que minutos de atividade elevada contribuem significativamente no consumo calórico.**

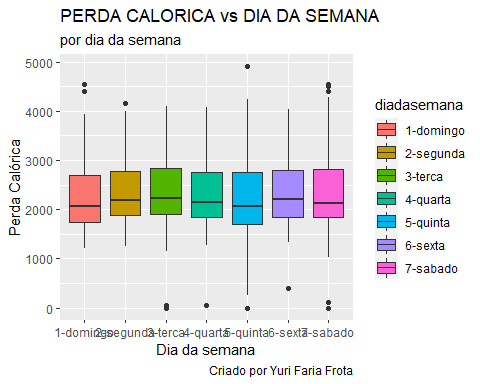
semanasono <- sqldf ('SELECT Id, SleepDay , TotalMinutesAsleep, CASE WHEN   
 SleepDay = "4/18/2016 12:00:00 AM" OR SleepDay = "4/25/2016 12:00:00 AM" OR SleepDay = "5/2/2016 12:00:00 AM" OR SleepDay = "5/9/2016 12:00:00 AM" THEN "2-segunda"  
 WHEN SleepDay = "4/12/2016 12:00:00 AM" OR SleepDay = "4/19/2016 12:00:00 AM" OR SleepDay = "4/26/2016 12:00:00 AM" OR SleepDay = "5/3/2016 12:00:00 AM" OR SleepDay = "5/10/2016 12:00:00 AM" THEN "3-terca"   
 WHEN SleepDay = "4/13/2016 12:00:00 AM" OR SleepDay = "4/20/2016 12:00:00 AM" OR SleepDay = "4/27/2016 12:00:00 AM" OR SleepDay = "5/4/2016 12:00:00 AM" OR SleepDay = "5/11/2016 12:00:00 AM" THEN "4-quarta"   
 WHEN SleepDay = "4/14/2016 12:00:00 AM" OR SleepDay = "4/21/2016 12:00:00 AM" OR SleepDay = "4/28/2016 12:00:00 AM" OR SleepDay = "5/5/2016 12:00:00 AM" OR SleepDay = "5/12/2016 12:00:00 AM" THEN "5-quinta"   
 WHEN SleepDay = "4/15/2016 12:00:00 AM" OR SleepDay = "4/22/2016 12:00:00 AM" OR SleepDay = "4/29/2016 12:00:00 AM" OR SleepDay = "5/6/2016 12:00:00 AM" THEN "6-sexta"   
 WHEN SleepDay = "4/16/2016 12:00:00 AM" OR SleepDay = "4/23/2016 12:00:00 AM" OR SleepDay = "4/30/2016 12:00:00 AM" OR SleepDay = "5/7/2016 12:00:00 AM" THEN "7-sabado"   
 WHEN SleepDay = "4/17/2016 12:00:00 AM" OR SleepDay = "4/24/2016 12:00:00 AM" OR SleepDay = "5/1/2016 12:00:00 AM" OR SleepDay = "5/8/2016 12:00:00 AM" THEN "1-domingo"   
 ELSE 0 end as diadasemana FROM "sleepDay\_merged" ')

ggplot(data=semana1) +  
 geom\_point(mapping=aes(x=TotalDistance, y= Calories, color = diadasemana ))+  
 labs (title = "PERDA CALORICA vs DISTANCIA TOTAL ", subtitle = "por dia da semana",   
 caption= "Criado por Yuri Faria Frota", x = "Distância Total (km)", y = "Perda Calórica")

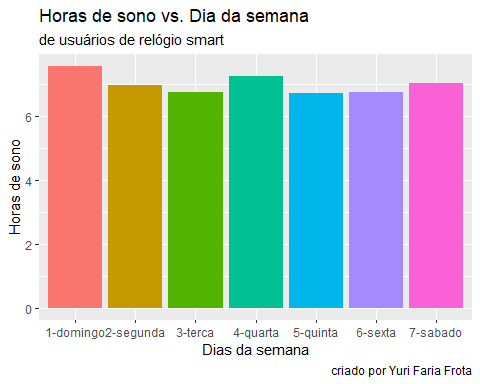


**Numa distribuição pelos dias da semana, podemos observar que não há nenhuma discrepância observável. Ou seja, os usuários não possuem habitos diferentes durante dias de semana ou finais de semana.**

ggplot(data=semana1) +  
 geom\_boxplot(mapping=aes(x=diadasemana, y= Calories, fill = diadasemana ))+  
 labs (title = "PERDA CALORICA vs DIA DA SEMANA ", subtitle = "por dia da semana",   
 caption= "Criado por Yuri Faria Frota", x = "Dia da semana", y = "Perda Calórica")

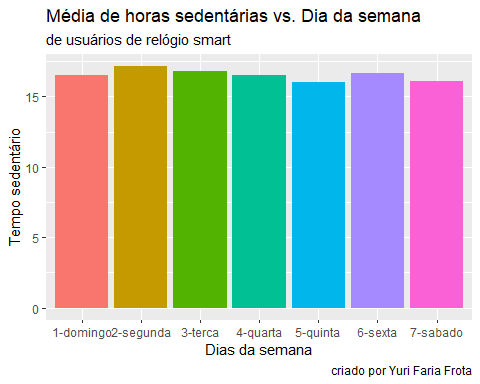


semanasono %>%   
 group\_by(diadasemana) %>%   
 summarise(totalsono = mean(TotalMinutesAsleep)) %>%   
 ggplot(aes(x = diadasemana, y = totalsono/60, fill = diadasemana)) +  
 geom\_col() +  
 labs (title = "Horas de sono vs. Dia da semana", subtitle = "de usuários de relógio smart ",   
 caption= "criado por Yuri Faria Frota", x = "Dias da semana", y = "Horas de sono") + theme(legend.position = "none")



**Podemos notar um ligeiro acrescimo nas horas de sono aos domingos, e às quartas-feiras.**

semana1 %>%   
 group\_by(diadasemana) %>%   
 summarise( meansedentaryminutes= mean(SedentaryMinutes) ) %>%   
 ggplot(aes(x = diadasemana, y = (meansedentaryminutes/60), fill = diadasemana)) +  
 geom\_col() +  
 labs (title = "Média de horas sedentárias vs. Dia da semana", subtitle = "de usuários de relógio smart",   
 caption= "criado por Yuri Faria Frota", x = "Dias da semana", y = "Tempo sedentário") + theme(legend.position = "none")



**O gráfico acima representa uma média de horas sedentárias, por dia da semana. Note que a diferença é bem pequena, mostrando que de uma forma geral, as pessoas se mantem em atividade sedentária durante todos os dias, inclusive nos finais de semanas.**

## Fase 5 - Compartilhar (Share)

As análises acima nos levam as seguintes conclusões:

Em média, os usuários desses equipamentos smart são sedentários (média maior que 16 horas por dia com atividades sedentárias) e dormem pouco (menos do que as 8 horas habituais). Isso ja nos leva a um perfil dos usuários: **Pessoas que precisam melhorar sua qualidade de vida.**

Através dos gráficos vimos a importância de se manter ativo, tambêm com atividades mais intensas, gerando acrescimos significativos na perda calórica.

Vimos tambem que o simples fato de caminhar (atividade leve), já é o suficiente para manter uma boa taxa de atividade, gerando tambem impactos significativos no consumo de calorias. O importante é se exercitar.

Sugestões

* Queremos melhorar a qualidade de vida de nossos usuários. Enfatizar as características mais importantes, e mostrar a evolução para o usuário atraves de um feedback dos dados, pode fazê-lo se comprometer mais com sua qualidade de vida, e consequentemente com os produtos e serviços da Belabbeat.
* Criação de um banco de dados dos equipamentos da Belabbeat, para direcionar os insights de uma forma mais eficiente.
* O gasto calórico praticamente não varia de acordo com os dias da semana, poderiamos colocar alguma forma de incentivar a realização de atividades através de ‘metas’ a serem cumpridas? algum tipo de “gameficação”?
* Dados como peso e características do sono podem gerar importantes informações. É importante a criação de maneiras mais eficazes de se obter esses dados, e pesquisar mais afundo os reais motivos por trás da dificuldade de obtê-los.