

Bellabeat — Empoderando as mulheres para desbloquear todo o seu potencial

Um caso de estudo sobre produtos IoT em Health Care

Por Bruno da Costa Calegari

Agosto de 2022

Sumário

Objetivo do Projeto	3
Introdução	3
Bellabeat.....	3
1ª Etapa: Perguntar	5
2ª Etapa: Preparar	6
3ª Etapa: Processar	7
4ª Etapa: Analisar	12
5ª Etapa: Compartilhar	18
6ª Etapa: Agir	22

Objetivo do Projeto

Introdução

Neste estudo de caso proposto pelo Google, iremos fazer uma análise para a empresa [Bellabeat](#), uma fabricante de produtos de alta tecnologia voltado à saúde das mulheres, para responder os problemas de negócios propostos vamos fazer uso das seis etapas da análise da dados: perguntar, preparar, processar, analisar, compartilhar e agir.

Bellabeat

Urška Sršen e Sando Mur fundaram a Bellabeat, uma empresa de alta tecnologia que fabrica produtos inteligentes focados na saúde das mulheres. Urška aproveitou sua experiência como artista para desenvolver uma tecnologia elegante que informa e inspira mulheres em todo o mundo. A coleta de dados sobre atividade, sono, estresse e saúde reprodutiva permitiu à Bellabeat capacitar as mulheres com conhecimento sobre sua própria saúde e hábitos. Desde que foi fundada em 2013, a Bellabeat cresceu rapidamente e não levou muito tempo para se consolidar como uma empresa de bem-estar voltada à tecnologia para mulheres.

Em 2016, a Bellabeat abriu escritórios ao redor do mundo e lançou vários produtos. Esses produtos tornaram-se disponíveis por meio de um número crescente de varejistas online, além do seu canal de comércio eletrônico em [seu site](#). A empresa investiu em mídia de publicidade tradicional, como rádio, outdoors, mídia impressa e televisão, mas se concentra principalmente no marketing digital. A Bellabeat investe ativamente na plataforma do Google, mantendo páginas ativas no Facebook e Instagram, além de engajar os consumidores de forma consistente no Twitter. Além disso, a Bellabeat exibe anúncios em vídeo no Youtube e anúncios gráficos na rede de display do Google para apoiar campanhas em datas importantes de marketing.

A Bellabeat é uma pequena empresa de sucesso, mas tem potencial para se tornar uma participante maior no mercado global de dispositivos inteligentes. Urška acredita que uma análise de dados de condicionamento físico de dispositivos inteligentes pode ajudar a abrir um leque de novas oportunidades de crescimento para a empresa.

Problema de negócio

Com um objetivo claro de identificar oportunidades potenciais de crescimento e recomendações para a melhoria da estratégia de marketing da Bellabeat, iremos nos basear nas tendências de uso de dispositivos inteligentes, foram geradas três perguntas principais para nortear a futura estratégia de marketing:

1. Quais são algumas das tendências no uso de dispositivos inteligentes?
2. Como essas tendências podem se aplicar aos clientes da Bellabeat?
3. Como essas tendências podem ajudar a influenciar a estratégia de marketing da Bellabeat?

Vamos analisar os dados de uso de dispositivos inteligentes (que não são da Bellabeat) para obter informações sobre como os consumidores usam esses dispositivos. Usaremos dados públicos que exploram os hábitos diários dos usuários de dispositivos inteligentes da **FitBit**:

- [Dados do rastreador de condicionamento físico FitBit](#) (CC0: Domínio público, conjunto de dados disponibilizado por [Mobius](#)): Este conjunto de dados do Kaggle, contém um rastreador de condicionamento físico pessoal de trinta usuários do Fitbit. Trinta usuários elegíveis do Fitbit consentiram com o envio de dados pessoais do rastreador, incluindo os resultados a cada minuto de atividade física, frequência cardíaca e monitoramento do sono. São abrangidas informações sobre atividades diárias, passos e frequência cardíaca que podem ser usadas para explorar os hábitos dos usuários.



1ª Etapa: Perguntar

A primeira fase da análise de dados consiste em definir o problema a ser resolvido, no caso da empresa Bellabeat, identificar oportunidades potenciais de crescimento e recomendações para a melhoria da estratégia de marketing.

Para guiar a nossa análise podemos formular questões usando o método SMART, esse método consiste em fazer perguntas:

- Específicas
- Mensuráveis
- Orientadas para a ação
- Relevantes
- Com duração definida.

Além do uso do método, deve se fazer perguntas abertas as partes interessadas, evitando perguntas sugestivas, fechadas ou vagas. Assim podemos entender melhor todo o contexto e ficar totalmente alinhado com as expectativas geradas.

Portanto, com a ajuda do método SMART, formulamos três perguntas essenciais para ser a alçada da nossa análise:

1. Quais são as tendências identificadas nos dados da FitBit?
2. Como essas tendências podem se aplicar aos usuários da Bellabeat?
3. Como usar essas tendências para formular uma nova estratégia de marketing para a Bellabeat?

A partir desses insights, a equipe de marketing criará uma nova estratégia. Portanto nossa análise deve ser apoiada com insights de dados convincentes e visualizações de dados objetivas e relevantes.

Como já temos nossa tarefa de negócio bem definida e alinhada com as partes interessadas, podemos prosseguir para a próxima fase da análise.



2ª Etapa: Preparar

Com o nosso problema de negócio bem definido, vamos preparar os dados para a exploração. Iremos usar dados públicos gerados pelos participantes de uma pesquisa distribuída pela Amazon Mechanical Turk entre 12 de março de 2016 e 12 de maio de 2016.

Esses dados possuem algumas limitações, como por exemplo:

- São dados de 2016, os dados podem ter mudado ao longo dos anos, sendo assim, podem não ser tão relevantes atualmente.
- O tamanho da amostra é de 30 pessoas, como não sabemos o sexo de cada participante, essa amostra pode não ser representativa de toda a população feminina.
- Como são dados provenientes de uma pesquisa, não podemos verificar totalmente a integridade ou precisão desses dados.

Vamos utilizar método ROAAC, **R**eval, **O**riginal, **A**brangente e **C**itado para identificar se temos uma boa fonte de dados:

- Real – **BAIXO** – Dados com baixa confiabilidade por conta do tamanho da amostra e da falta de informação sobre o sexo dos participantes.
- Original – **BAIXO** – Fornecedor terceirizado.
- Abrangente – **MÉDIO** – Os parâmetros dos produtos da FitBit se assemelham aos produtos da Bellabeat.
- Atual – **BAIXO** – Não são dados atuais, já possuem 6 anos.
- Citado – **MÉDIO** – Dados coletados por terceiros, porém de uma fonte com alta confiabilidade.

Nosso conjunto de dados pode ser classificado como de má qualidade, sendo assim, não é recomendável produzir recomendações para tomada de decisões nos negócios com base nesses dados, porém como estamos fazendo apenas um estudo de caso, iremos prosseguir com esse conjunto de dados.



3ª Etapa: Processar

Nessa etapa, para a preparação e processamento dos dados, usaremos a linguagem R.

1. Preparando o ambiente

- Instalando e carregando o pacote tidyverse, um pacote projetado para a ciência de dados, para mais informações sobre o pacote acesse o site do [tidyverse](https://www.tidyverse.org/).

```
install.packages("tidyverse")  
library(tidyverse)
```

2. Importando nosso conjunto de dados

- Definindo nossa área de trabalho, foi substituído o real caminho do arquivo por “...”.

```
setwd("C:\\...\\Fitabase Data 4.12.16-5.12.16")
```

- Usaremos a função ‘read_csv’ para importar nosso conjunto de dados e atribuiremos cada conjunto a uma variável.

```
activity <- read.csv("dailyActivity_merged.csv")  
intensities <- read.csv("hourlyIntensities_merged.csv")
```

3. Fazendo a limpeza e manipulação dos nossos dados

- Vamos primeiro nos familiarizar com nossos dois conjuntos de dados, executaremos essas quatro funções para cada um dos conjuntos:

```
head()  
str()  
glimpse()  
summary()
```


- Activity:

A função `head()` nos traz as seis primeiras linhas de todas as colunas contida no conjunto de dados:

```
> head(activity)
  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 FairlyActiveMinutes
1                6.06                  0              25              13
2                4.71                  0              21              19
3                3.91                  0              30              11
4                2.83                  0              29              34
5                5.04                  0              36              10
6                2.51                  0              38              20
  LightlyActiveMinutes SedentaryMinutes Calories
1                328              728      1985
2                217              776      1797
3                181             1218      1776
4                209              726      1745
5                221              773      1863
6                164              539      1728
```

A função `str()` nos traz a quantidade de observações(linhas) e de variáveis(colunas) e seus respectivos nomes e formatações, como alguns de seus valores também:

```
> str(activity)
'data.frame': 940 obs. of 15 variables:
 $ Id : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
 $ ActivityDate : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
 $ TotalSteps : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
 $ TotalDistance : num  8.5 6.97 6.74 6.28 8.16 ...
 $ TrackerDistance : num  8.5 6.97 6.74 6.28 8.16 ...
 $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
 $ VeryActiveDistance : num  1.88 1.57 2.44 2.14 2.71 ...
 $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
 $ LightActiveDistance : num  6.06 4.71 3.91 2.83 5.04 ...
 $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
 $ VeryActiveMinutes : int  25 21 30 29 36 38 42 50 28 19 ...
 $ FairlyActiveMinutes : int  13 19 11 34 10 20 16 31 12 8 ...
 $ LightlyActiveMinutes : int  328 217 181 209 221 164 233 264 205 211 ...
 $ SedentaryMinutes : int  728 776 1218 726 773 539 1149 775 818 838 ...
 $ Calories : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

A função *glimpse()* nos traz praticamente a mesma coisa que a função *str()*, porém com algumas pequenas diferenças:

```
> glimpse(activity)
Rows: 940
Columns: 15
$ Id <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 15...
$ ActivityDate <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016"...
$ TotalSteps <int> 13162, 10735, 10460, 9762, 12669, 9705, 13019, 155...
$ TotalDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6...
$ TrackerDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6...
$ LoggedActivitiesDistance <dbl> 0, 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, 3.53, 1...
$ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.32, 0...
$ LightActiveDistance <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.03, 4...
$ SedentaryActiveDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
$ VeryActiveMinutes <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 41, 39...
$ FairlyActiveMinutes <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21, 5, ...
$ LightlyActiveMinutes <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, 211, ...
$ SedentaryMinutes <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818, 838...
$ calories <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2035, 17...
```

A função *summary()* nos traz um resumo estatístico de todas as colunas contidas no conjunto de dados:

```
> summary(activity)
      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
```

head()

```
> head(intensities)
```

	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

$$str()$$

```
> str(Intensities)
'data.frame': 22099 obs. of 4 variables:
 $ Id : num 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
 $ ActivityHour : chr "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/2016 3:00:00 AM" ...
 $ TotalIntensity : int 20 8 7 0 0 0 0 13 30 ...
 $ AverageIntensity: num 0.333 0.133 0.117 0 0 ...
```

glimpse()

```
> glimpse(intensities)
Rows: 22,099
Columns: 4
$ Id          <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366...
$ ActivityHour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/201...
$ TotalIntensity <int> 20, 8, 7, 0, 0, 0, 0, 0, 13, 30, 29, 12, 11, 6, 36, 58, 13...
$ AverageIntensity <dbl> 0.333333, 0.133333, 0.116667, 0.000000, 0.000000, 0.000000...
```

summary()

```
> summary(intensities)
```

	Id	ActivityHour	TotalIntensity	AverageIntensity
Min.	:1.504e+09	Length:22099	Min. : 0.00	Min. :0.0000
1st Qu.	:2.320e+09	Class :character	1st Qu.: 0.00	1st Qu.:0.0000
Median	:4.445e+09	Mode :character	Median : 3.00	Median :0.0500
Mean	:4.848e+09		Mean : 12.04	Mean :0.2006
3rd Qu.	:6.962e+09		3rd Qu.: 16.00	3rd Qu.:0.2667
Max.	:8.878e+09		Max. :180.00	Max. :3.0000

Através dessas funções já podemos ver que não há nenhum valor nulo e nenhum valor faltando em nossos conjuntos de dados, agora vamos conferir a coluna *id* em todos nossos conjuntos para confirmar se existem 30 valores únicos, no caso, dos 30 participantes:

```
n distinct(activity$Id)
```

[1] 33

```
n distinct(intensities$Id)
```

[1] 33

Podemos ver que para o conjunto ‘*activity*’ e ‘*intensities*’ existem trinta e três participantes, três participantes a mais do que o esperado.

Vamos tratar a coluna ‘*ActivityHour*’ do conjunto ‘*intensities*’, iremos converter os formatos ‘m/d/y h:m:s am/pm’ e separaremos as datas das horas:

```
intensities$ActivityHour <- as.POSIXct(intensities$ActivityHour, format =
"%m/%d/%Y %I:%M:%S %p")

intensities %>%
  separate(
    col = ActivityHour,
    into = c("ActivityDate", "ActivityHour"),
    sep = " ") -> intensities
```

Assim criamos duas colunas, uma para guardar a data e outra apenas para as horas, porém a coluna de data está no formato “*chr*” nos nossos dois conjuntos de dados, vamos fazer a conversão para ‘*date*’:

```
activity$ActivityDate <- as.Date(activity$ActivityDate, format =
"%m/%d/%Y")

intensities$ActivityDate <- as.Date(intensities$ActivityDate, format =
"%m/%d/%Y")
```

Para nosso conjunto de dados ‘*activity*’ vamos criar uma nova coluna chamada ‘*weekday*’ extraindo esses dados da coluna ‘*ActivityDate*’ e formatando nossos dados como fatores:

```
activity$weekday <- weekdays(activity$ActivityDate)
activity$weekday <- as.factor(activity$weekday)
```

Também vamos criar um novo ‘*dataframe*’ a partir desse mesmo conjunto de dados, nesse novo ‘*dataframe*’ iremos incluir as colunas ‘*VeryActiveMinutes*’, ‘*FairlyActiveMinutes*’, ‘*LightlyActiveMinutes*’, ‘*SedentaryMinutes*’ e suas respectivas somas para uma análise posterior:

```
ActiveMinutes <- c("VeryActiveMinutes", "FairlyActiveMinutes",
"LightlyActiveMinutes", "SedentaryMinutes")

Total <- c(sum(activity$VeryActiveMinutes),
sum(activity$FairlyActiveMinutes), sum(activity$LightlyActiveMinutes),
sum(activity$SedentaryMinutes))

TotalActiveMinutes <- data.frame(ActiveMinutes, Total)
```

Com os nossos dados devidamente tratados e limpos podemos prosseguir para a fase de análise.

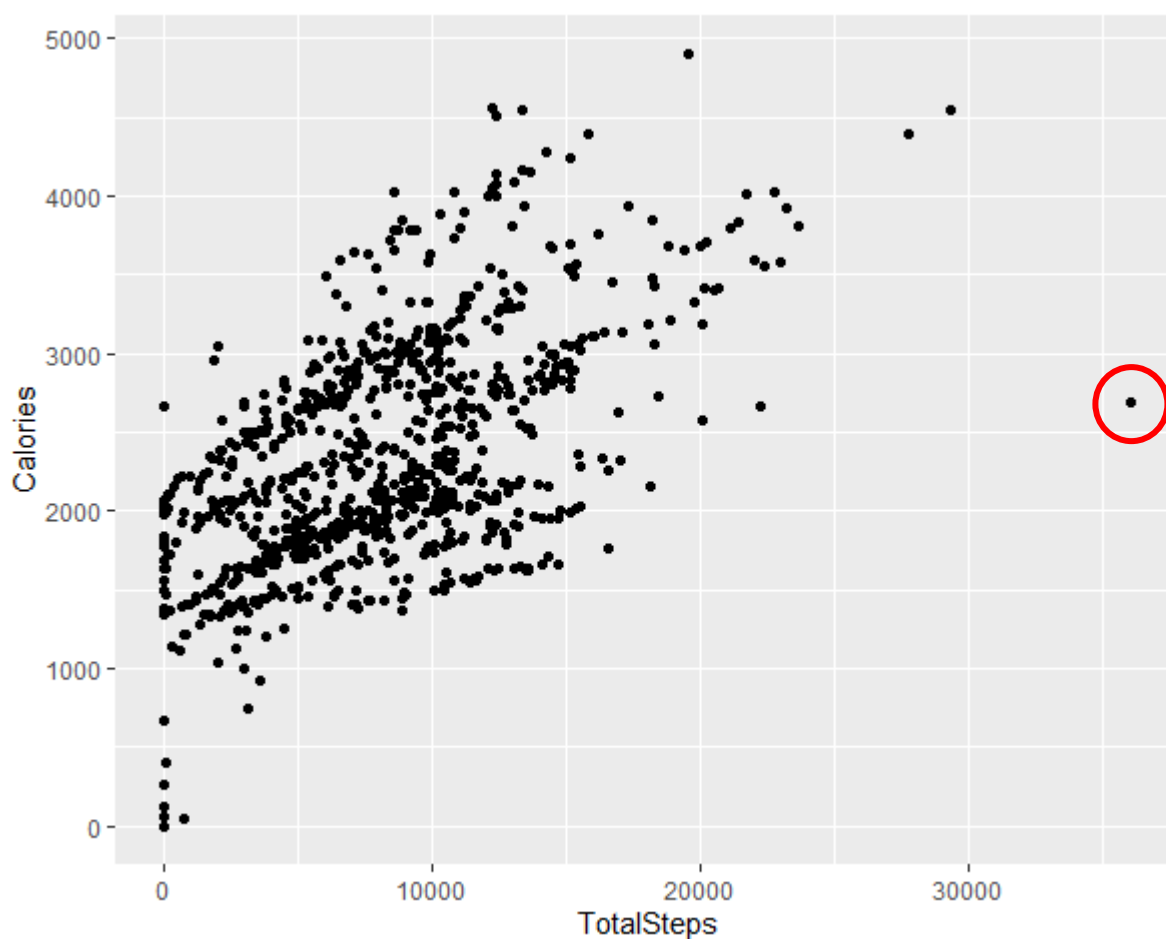


4ª Etapa: Analisar

Com os dados tratados e limpos, podemos olhar e explorar nosso conjunto de dados para fazer conexões e identificar tendências:

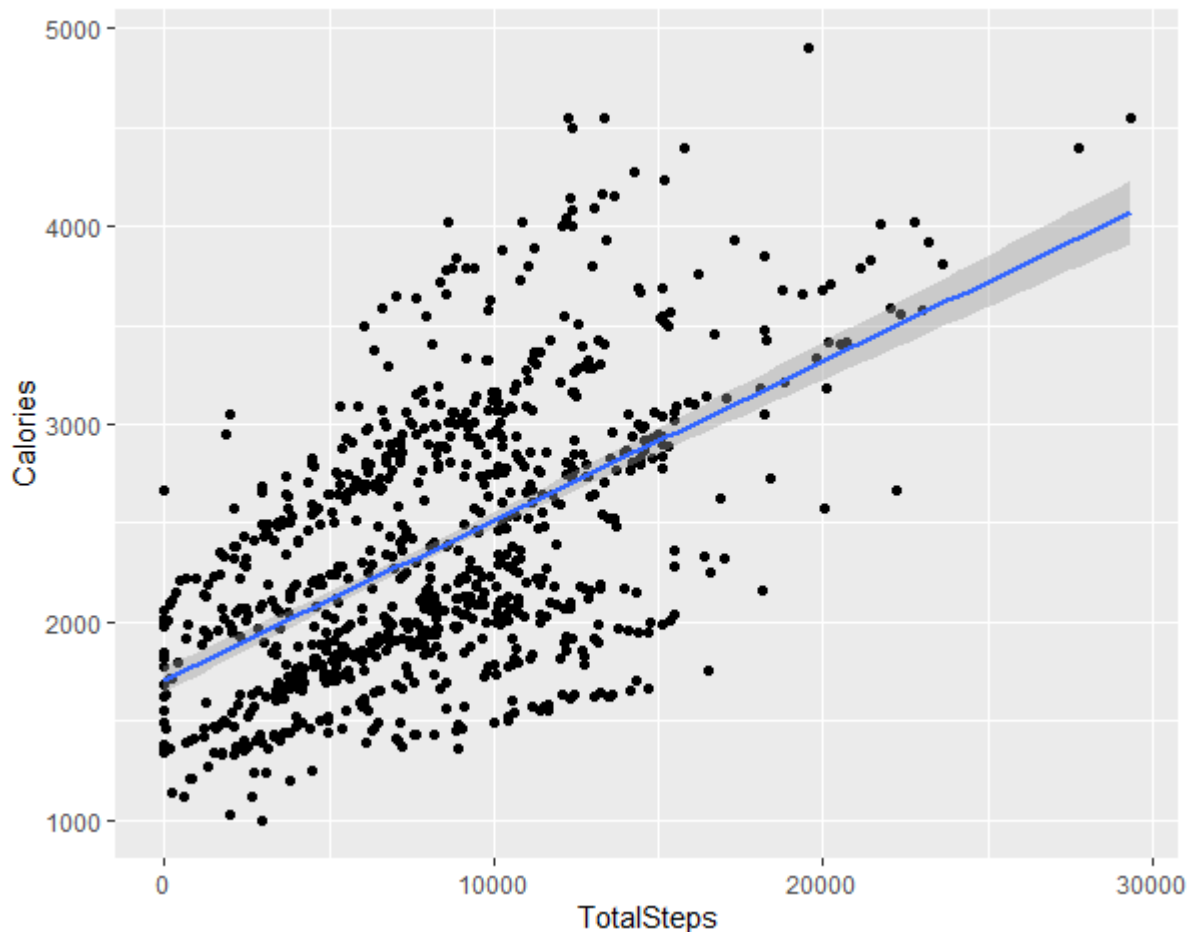
- Começaremos olhando a correlação entre o total de passos e a quantidade de calorias gastas:

```
ggplot(activity) +  
  aes(TotalSteps, Calories) +  
  geom_point()
```



Podemos ver uma correlação positiva para nossas duas variáveis, para melhorar nossa visualização, vamos deletar retirar o outlier circulado em vermelho e adicionar uma linha de tendência de modelo linear:

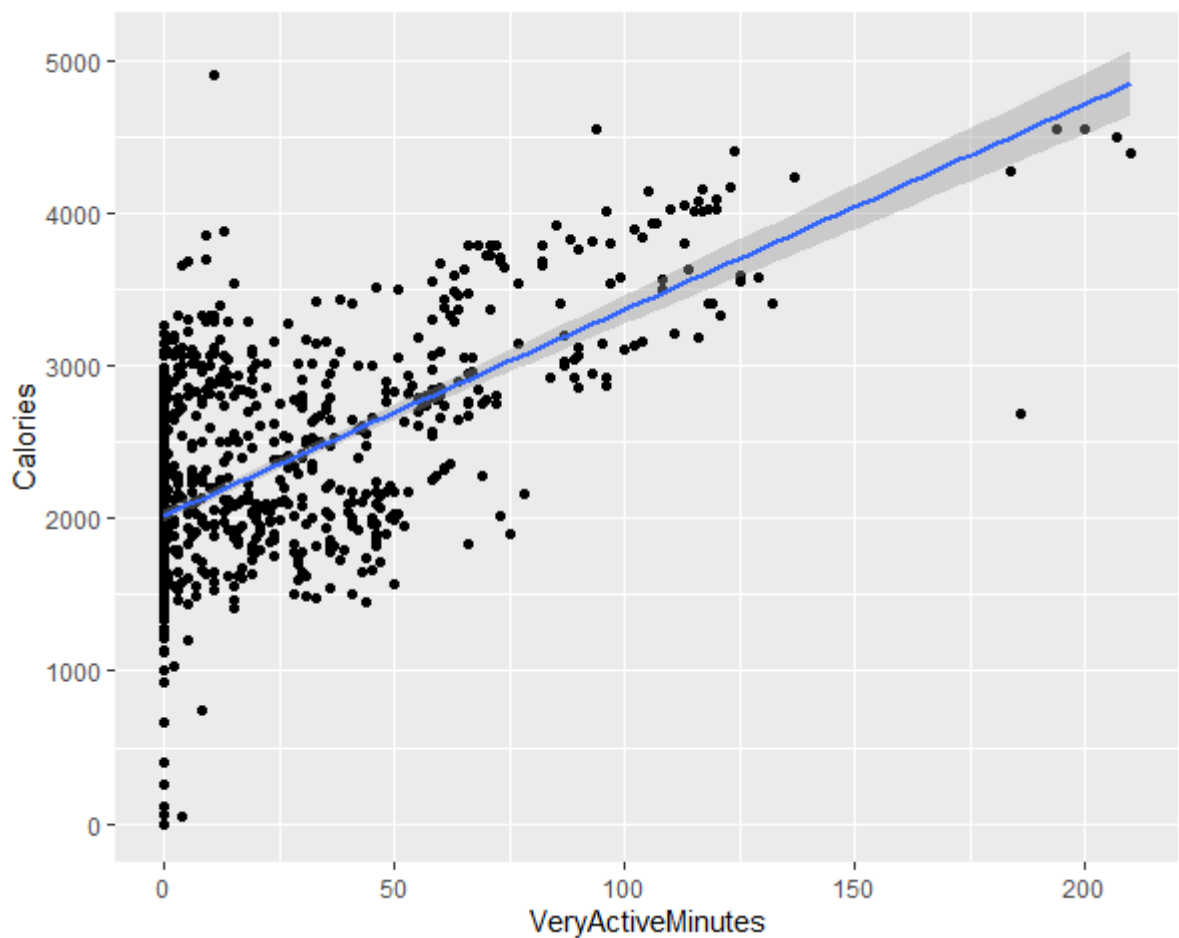
```
activity %>%
  filter(TotalSteps < 30000 & Calories > 1000) %>%
  ggplot(aes(TotalSteps, Calories)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Quanto mais ativos são os usuários, mais calorias eles gastam. São gastas em média 2304 calorias, enquanto que, são dados em média 7638 passos. Também podemos ver que até os primeiros 15.000 passos a taxa de calorias gastas aumentam de forma intensa, esfriando lentamente após essa faixa.

Agora, analisaremos se os minutos onde os usuários estão mais ativos também possui uma relação com o número de calorias gastas:

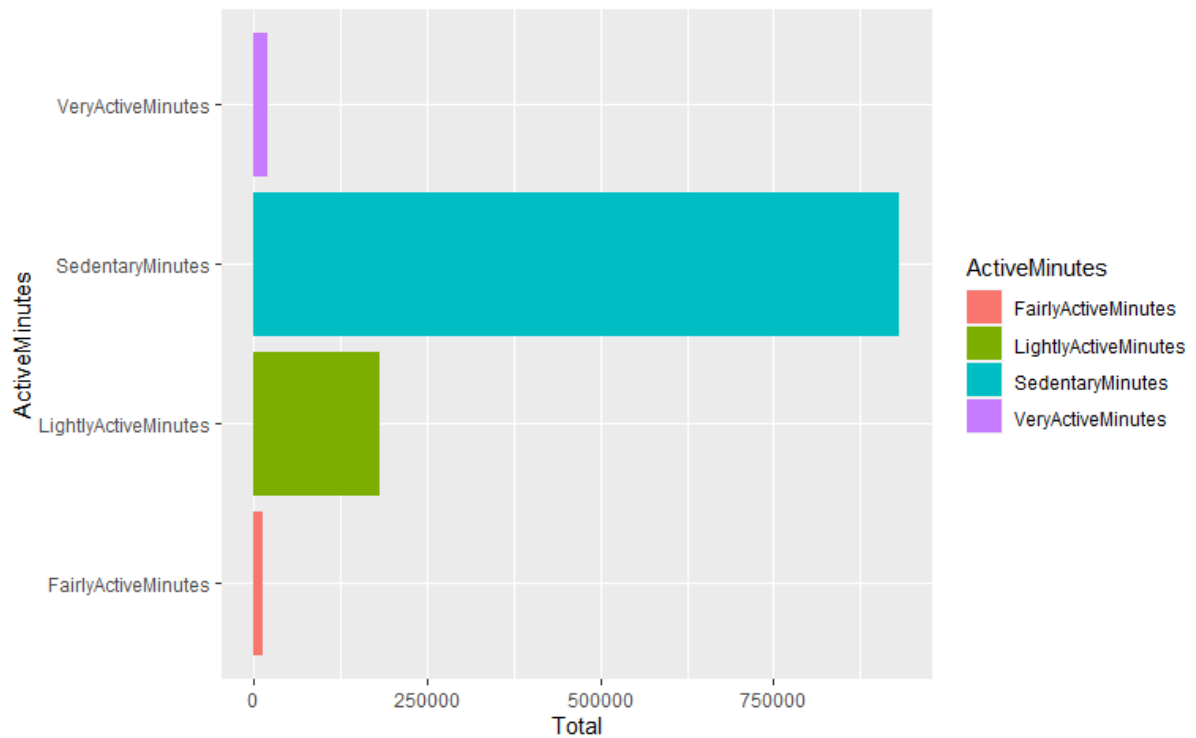
```
ggplot(activity) +
  aes(VeryActiveMinutes, Calories) +
  geom_point() +
  geom_smooth(method = 'lm')
```



E a resposta é sim, obviamente, quanto mais minutos de atividade alta, como uma corrida por exemplo, mais calorias serão gastas pelo usuário.

Agora usaremos o nosso *'dataframe'* criado para gerar um gráfico de barras, veremos se os usuários estão tendo mais minutos de atividade alta, média, baixa, ou de tempo sedentário (parado):

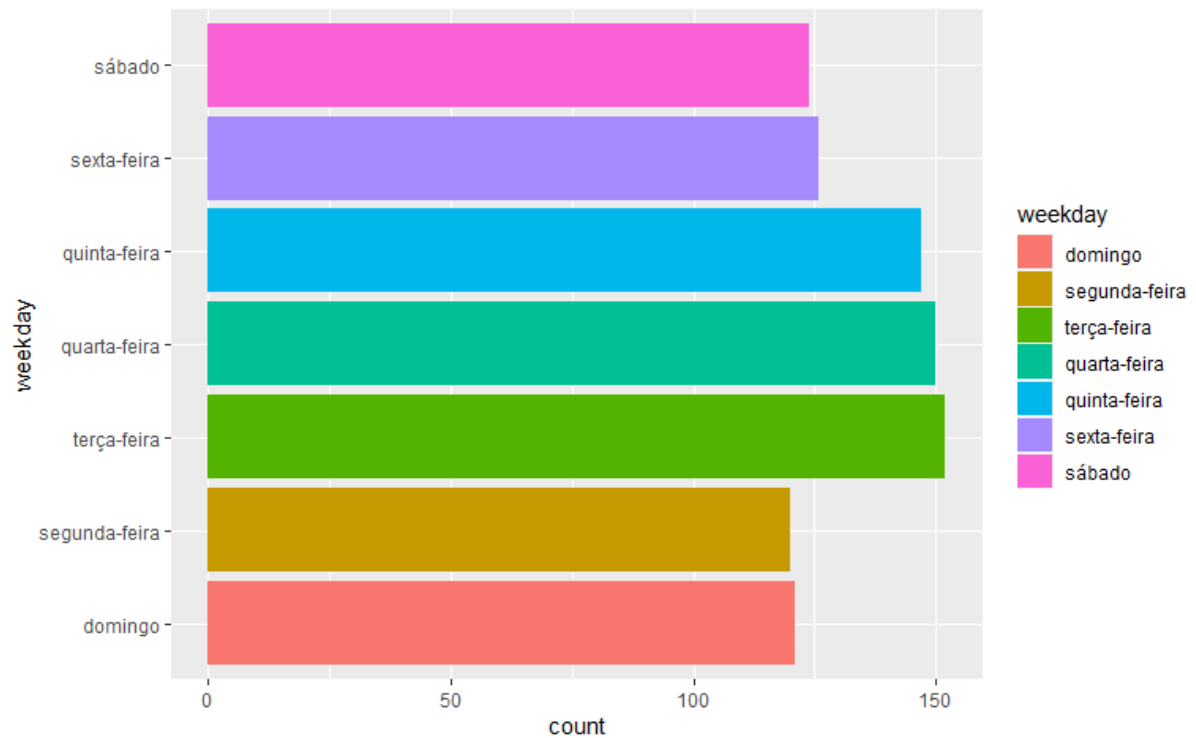
```
ggplot(TotalActiveMinutes) +  
  aes(ActiveMinutes, Total, fill = ActiveMinutes) +  
  geom_bar(stat = "Identity") +  
  coord_flip()
```



Podemos constatar que o tempo sedentário, ou tempo parado é o mais alto, o tempo de pouca atividade (como por exemplo uma caminhada) é mais alto que o tempo de média ou alta atividade (como uma corrida por exemplo). Vemos que os participantes dessa pesquisa não costumam praticar atividades físicas com frequência.

A seguir, iremos dar uma olhada na frequência semanal de uso do aplicativo pelos usuários:

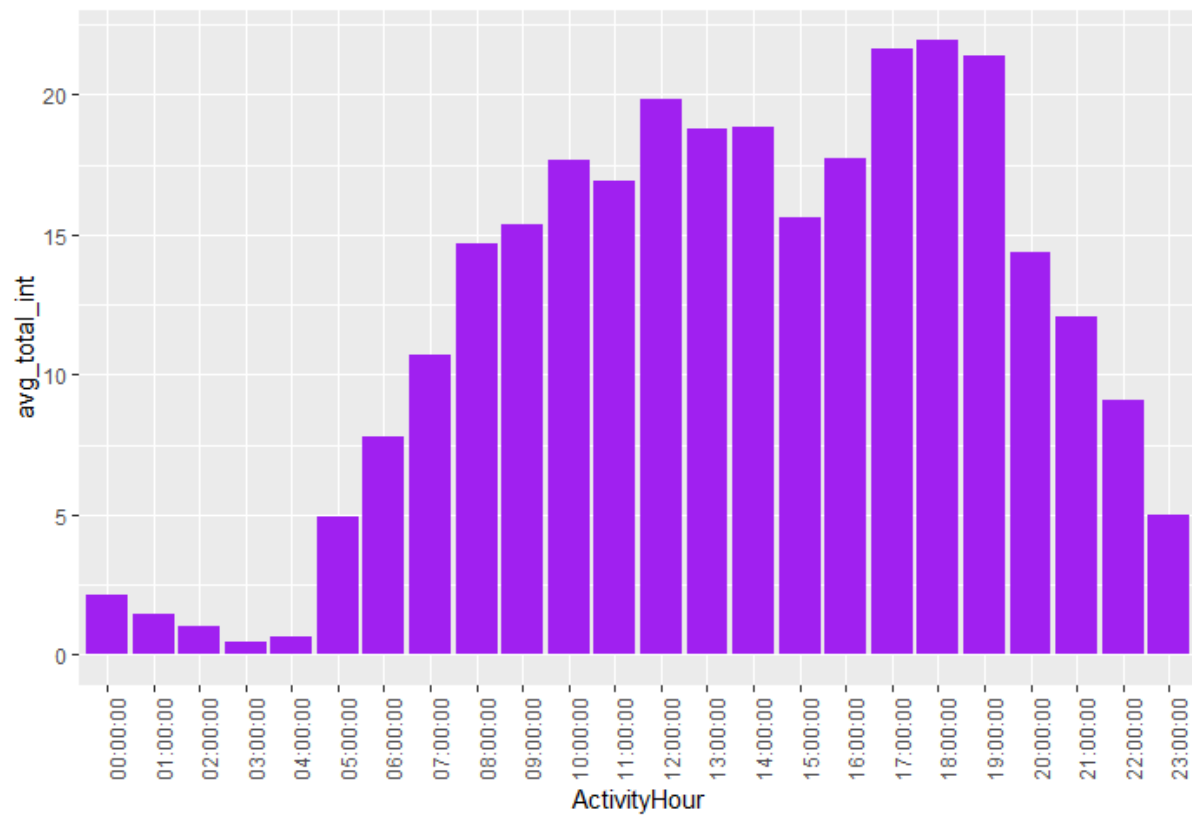
```
activity %>%
  mutate(weekday = factor(weekday, c("domingo", "segunda-feira", "terça-
feira", "quarta-feira", "quinta-feira", "sexta-feira", "sábado"))) %>%
  ggplot(aes(weekday, fill = weekday)) +
  geom_bar() +
  coord_flip()
```

Vemos que o maior uso acontece no meio da semana, aumentando a partir de terça-feira e começando a diminuir a partir de quinta-feira.

Já analisamos nosso primeiro conjunto de dados, vamos dar uma olhada agora no conjunto *'intensities'*. Procedendo, vamos gerar um histograma para olhar a média de intensidade durante o uso do app ao longo do dia:

```
intensities %>%
  group_by(ActivityHour) %>%
  summarise(avg_total_int = mean(TotalIntensity)) %>%
  ggplot(aes(ActivityHour, avg_total_int,)) +
  geom_histogram(stat = "identity", fill = "purple") +
  theme(axis.text.x = element_text(angle = 90))
```



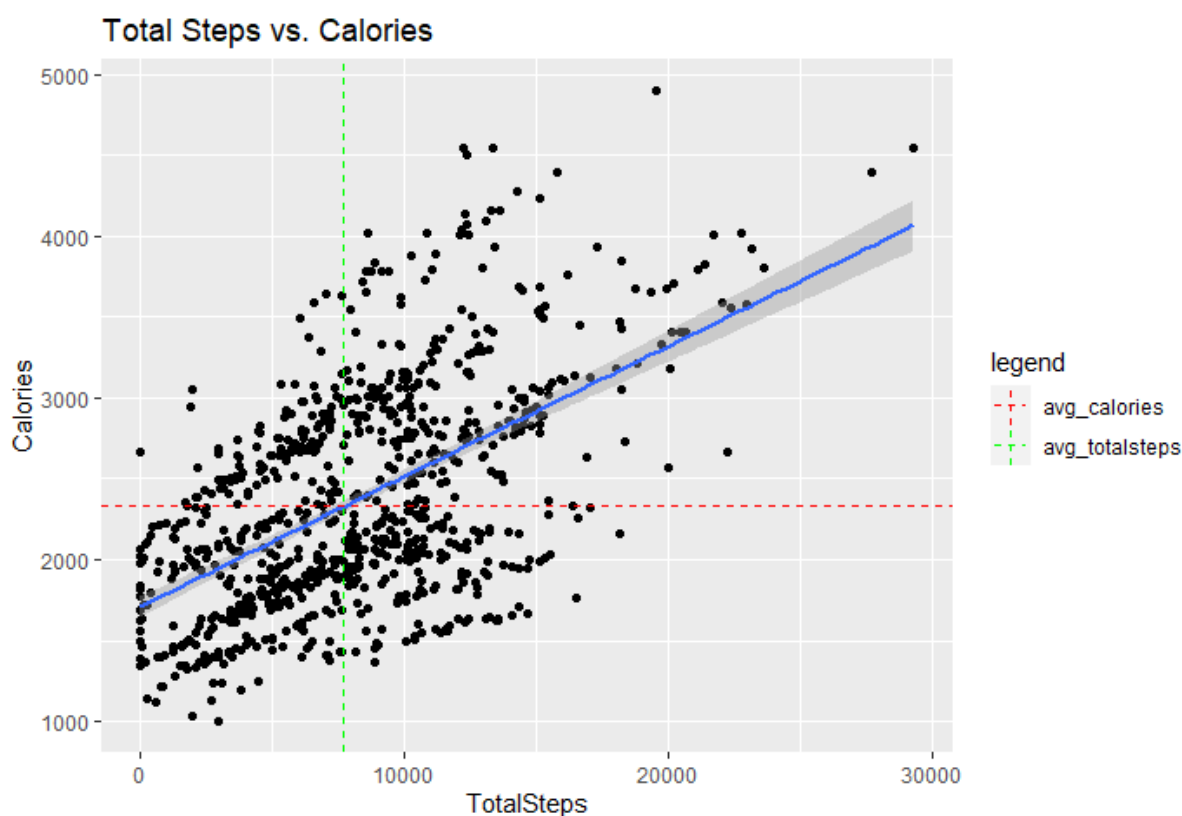
Vemos que as pessoas são mais ativas entre as 5h às 22h, com uma frequência mais intensa entre as 17h às 19h, provavelmente possa ser o horário que as pessoas saem do trabalho e vão fazer uma caminhada, ou malhar.



5ª Etapa: Compartilhar

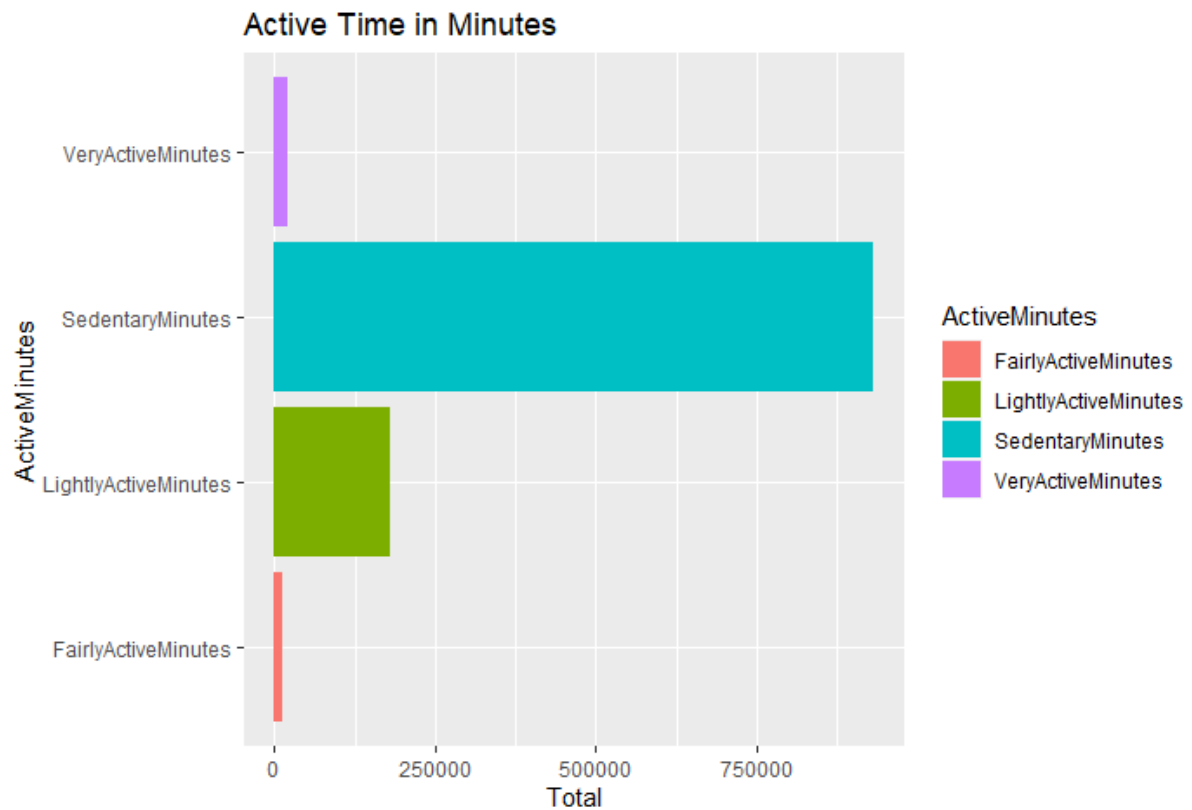
Após realizar a análise e ver o que os dados nos contam, vamos compartilhar nossas principais conclusões com visualizações polidas e objetivas.

1. Total de passos para quantidade de calorias gastas:



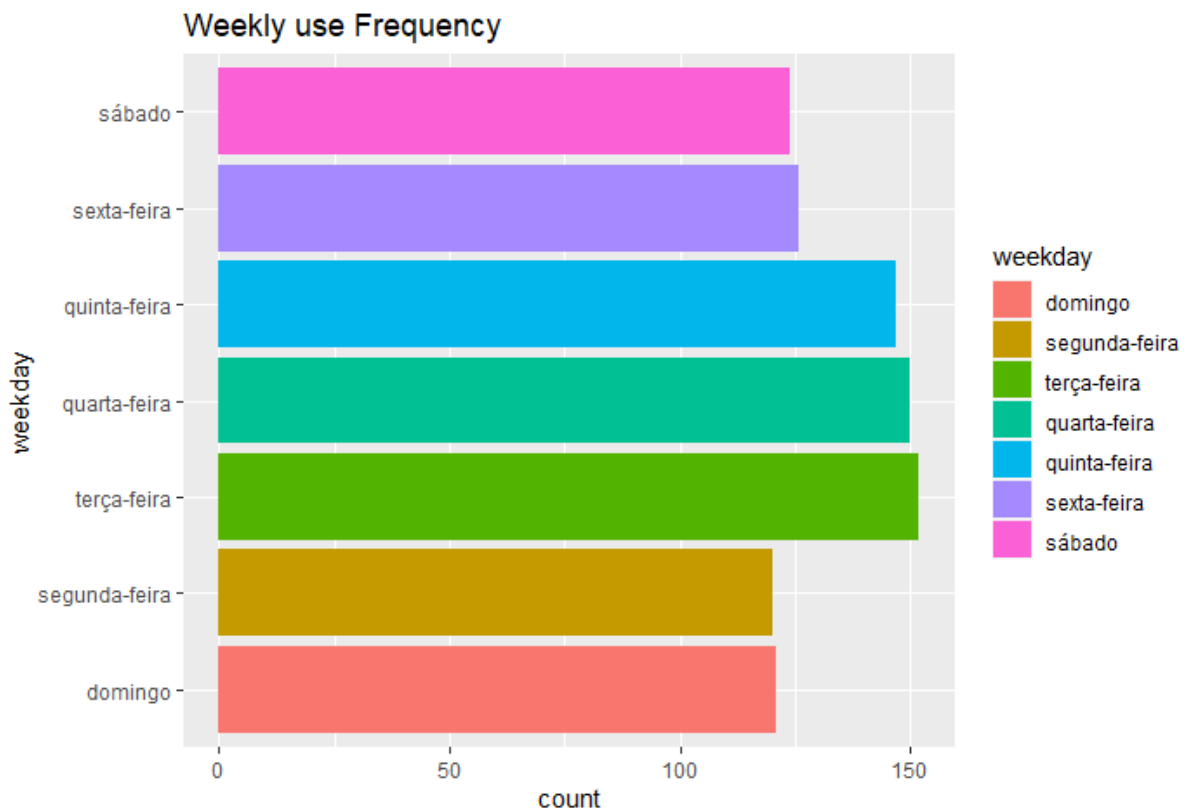
- ✓ A média do total de passos por dia é de 7638 e a média de calorias gastas por dia são de 2304.
- ✓ Pesquisas como a da [Harvard Medical School](#) dizem que as taxas de mortalidade diminuam progressivamente antes dos 7.500 passos por dia aproximadamente, já começamos a ver benefícios significativos a partir dos 4.400 passos/d, portanto a média total de passos está acima da média proposta pelas pesquisas mais atuais.

2. Tempo de atividade em minutos:



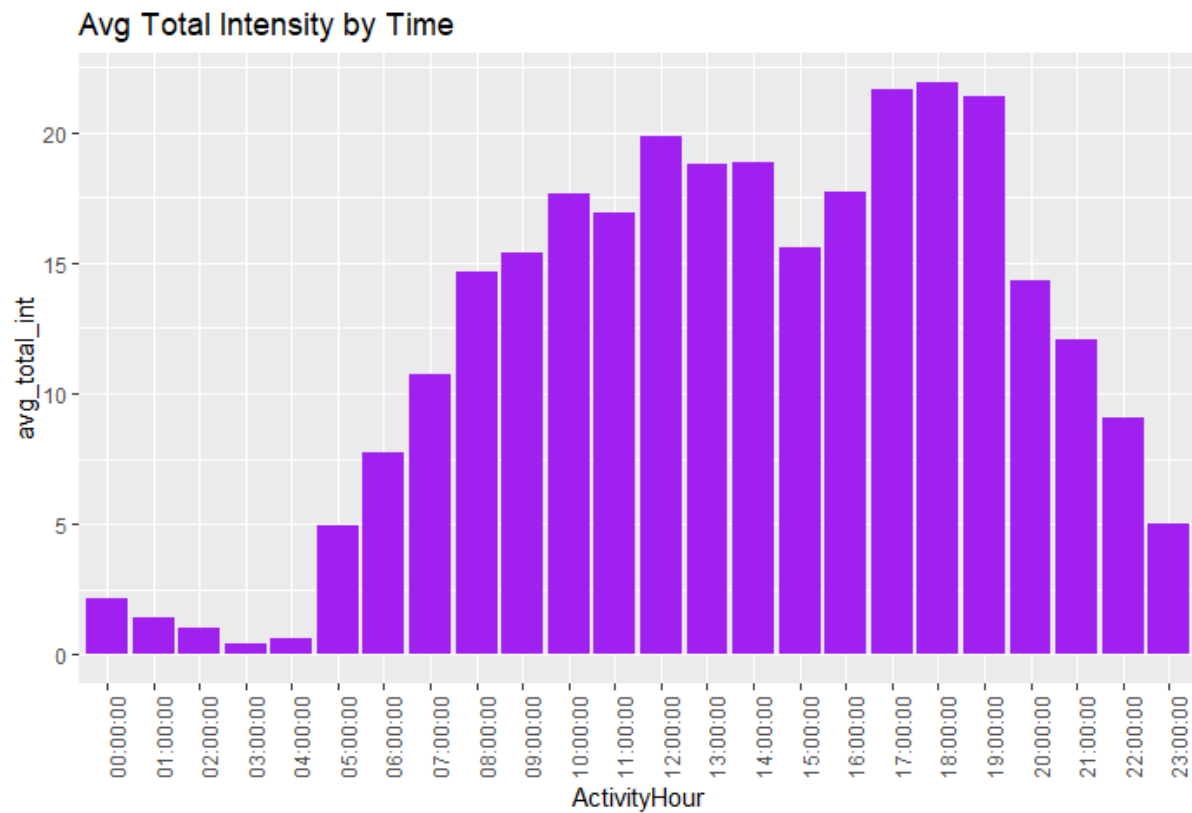
- ✓ O tempo de sedentarismo, ou tempo parado, corresponde a 81% do tempo de uso do app.
- ✓ O aplicativo raramente está sendo usado para rastrear atividades físicas, de acordo com as porcentagens as atividades de alta ou média intensidade não chegam a somar nem 5%. Visto que o aplicativo FitBit foi desenvolvido para incentivar o condicionamento físico, são dados desanimadores.

3. Frequência de uso semanal:



- ✓ Os dias com maior frequência de uso do aplicativo são durante a semana, de terça a quinta-feira.
- ✓ Os usuários podem realmente usarem o aplicativo com mais frequência durante o meio da semana, ou apenas estarem se lembrando de rastrear suas atividades físicas durante esses dias.

4. Média de Intensidade Total por Tempo:



- ✓ Usuários são mais ativos das 5h às 22h.
- ✓ Os horários de maior frequência de atividade intensa são das 17h às 19h, provavelmente pois são os horários que as pessoas saem do trabalho e vão fazer uma caminhada, ou vão para a academia malhar.



6ª Etapa: Agir

Na última etapa do processo de análise de dados, forneceremos nossas principais recomendações com base em nossas descobertas, sendo assim, vamos responder as nossas três perguntas formuladas anteriormente:

1. Quais são as tendências identificadas nos dados da FitBit?
 - ✓ A média de passos totais por dia já está acima da média proposta pelas mais novas pesquisas.
 - ✓ Os usuários costumam rastrear mais suas atividades durante os dias de semana, com uma frequência bem alta no meio da semana, de terça a quinta-feira.
 - ✓ As pessoas costumam ser mais ativas das 5h às 22h, com uma maior frequência de intensidade das 17h às 19h.
2. Como essas tendências podem se aplicar aos usuários da Bellabeat?
 - ✓ Ambas as empresas desenvolvem produtos semelhantes, focados em fornecer às mulheres dados de sua saúde e incentivá-las a tomar decisões saudáveis. Essas tendências em comum, giram em torno da saúde e podem ser aplicadas aos clientes da Bellabeat.
3. Como usar essas tendências para formular uma nova estratégia de marketing para a Bellabeat?

Vamos dividir essa pergunta em três respostas, um possível público alvo, uma mensagem para a companhia sobre sua missão e algumas ideias para o aplicativo da Bellabeat com base nessas tendências.

I. Público Alvo:

- Pessoas que trabalham em tempo integral, ou passam muito tempo no computador, ou escritório. Os usuários já costumam fazer algumas atividades leves para se manterem saudáveis. Para melhorar essas atividades e manter esses hábitos saudáveis é interessante um aplicativo para disponibilizar informações sobre sua saúde e hábitos, e também maneiras para desenvolver hábitos saudáveis e motivação para mantê-los.

II. Mensagem para a companhia:

- O aplicativo da Bellabeat precisa ter uma missão bem definida, um exemplo seria: ser um aplicativo exclusivo para atividade física. Sendo um guia para seus usuários e clientes para ajudar a equilibrar a vida pessoal e profissional mantendo hábitos saudáveis.

III. Ideias para o aplicativo:

- Explicar os benefícios para a saúde de dar mais de 4.500 passos por dia e encorajar seus usuários com notificações e mensagens motivacionais, também adicionar mensagens de reconhecimento para usuários que passem da marca de 5.000 ou 7.500 passos por dia.
- Usar notificações para avisar os usuários da hora de dormir e dos benefícios de um bom sono, como também adicionar um controle de calorias no aplicativo e sugerir ideias de refeições de baixa caloria para usuários que desejam perder peso.
- Aos finais de semana, mandar notificações lembrando e encorajando os usuários a praticarem exercícios físicos.
- Programar mais notificações para os usuários com maior índice de tempo de sedentarismo, com mensagens encorajadoras, lembretes e informações sobre os benefícios para a saúde de se manter hábitos saudáveis.
- Programar mais notificações para os horários de pico, das 17h às 19h, lembrando os usuários de irem fazer uma caminhada, ou irem para a academia malhar.