Case study - Bellabeat app

Douglas Silva Alves

2024-01-04

About this document

Welcome! This document represents the culmination of my efforts as part of the final project for the Google Data Analytics Professional Certificate. In this captivating case study, I step into the role of a junior data analyst tasked with a project for the esteemed company 'Bellabeat'.

Project Overview

Business task: Analyzing smart device usage data to extract valuable insights into consumer behaviors regarding non-Bellabeat smart devices and recommendations for the Bellabeat marketing strategy improvement.

Key Questions for Analysis:

1 Smart Device Usage Trends: Uncover and understand prevailing trends in smart device usage.

2 Applicability to Bellabeat Customers: Assess how these identified trends can be applied to enhance the experience for Bellabeat customers.

3 Impact on Marketing Strategy: Determine how these trends can play a pivotal role in shaping and influencing Bellabeat's marketing strategy.

Embark on this journey with me as I delve into the intricate world of data analysis, providing strategic insights that have the potential to reshape the landscape for Bellabeat. Let's explore the realm of possibilities together!

Data for this project

The data used for this project is an open access data available here

Loading packages

```
library("tidyverse")
library("dplyr")
library("magrittr")
library("qgraph")
```

Loading datasets

```
daily_sleep_data <- read.csv("data/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv")
daily_activity_data <- read.csv("data/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
```

Ensuring Successful Dataset Loading with the Kable Function

Join me in a quick check to verify the seamless loading of our datasets using the versatile kable function. This step is crucial to establish a robust foundation for our analysis. Are the datasets ready to reveal their secrets?

```
knitr::kable(head(daily_sleep_data), format="markdown")
```

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

Assessing Dataset Structure and Date Column Transformation

In our exploration of datasets using the str() function, we identified that the date column was initially in string format. However, this has already been successfully addressed through the application of the 'as.Date()' function (described below).

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

Join me in examining the refined dataset structure, now equipped with a more meaningful date format for our in-depth analysis.

Evaluating Data Collection: Monitoring Times Across Individuals

Let's delve into the intricacies of data collection by examining the distribution of data points for each individual. Notably, we observe varying monitoring times across individuals, shedding light on the temporal nuances within our dataset.

Examining the Number of Individuals in Each Dataset

Join me in a quick assessment to determine the count of individuals within each dataset. This simple yet crucial step will provide clarity on the composition of our data.

```
individuals_sleep_data <- daily_sleep_data %>% distinct(Id) #24 individuals
individuals_activity_data <- daily_activity_data %>% distinct(Id) #33 individuals
```

Merging Datasets: Aligning Date Columns

Our next step involves the seamless merging of the two datasets. To achieve this cohesion, it's imperative that both datasets share the same name for the date column. Join me in this process of aligning the date columns, paving the way for a unified and consolidated dataset. Let's ensure a harmonious combination of our data for a more comprehensive analysis.

```
colnames(daily_activity_data)[colnames(daily_activity_data) == "ActivityDate"] ="Date"
colnames(daily_sleep_data)[colnames(daily_sleep_data) == "SleepDay"] ="Date"
```

Dataset Fusion: Merging with ID and Date

Synthesize datasets by merging them based on common ID and date columns, forming the foundation for a comprehensive analysis.

```
merged_data <- daily_activity_data %>%
  left_join(daily_sleep_data, by = c('Id', 'Date'))
```

Data Organization: Sorting by Date

With our merged dataset comprising 33 individuals, let's ensure data cleanliness. Our first step involves organizing the data by date, arranging it from the oldest to the newest for each ID. This systematic arrangement sets the stage for a meticulous data quality assessment.

```
merged_data <- merged_data %>% arrange(Id, Date)
```

Introducing 'Week' Variable: Monitoring Duration

In our quest for comprehensive insights, let's enhance our dataset by creating a new variable named 'week'. This variable will signify the number of weeks each individual was monitored. Join me in this crucial step to add a temporal dimension to our analysis.

```
min_week <- min(isoweek(merged_data$Date))
merged_data <- merged_data %>%
  mutate(week = isoweek(Date) - min_week + 1)
```

Computing Weekly Metrics: Steps, Active Minutes, and More

Upon a brief inspection, it's apparent that individuals were monitored over 5 weeks, with variations in the number of monitoring days. Notably, discrepancies exist, such as in the last week, where some individuals logged 4 days and others 3. Our next step involves computing weekly metrics, including steps, very active minutes, fairly active minutes, lightly active minutes, sedentary minutes, calories, and total time in bed. These computations will serve as the foundation for our subsequent analysis on how these variables interrelate. Join me in estimating the intricate patterns within our dataset, laying the groundwork for further exploration.

```
weekly_merged_data <- merged_data %>%
  group_by(Id, week) %>%
  summarise(
    wk_steps = sum(TotalSteps),
```

```
    very_activemin = sum(VeryActiveMinutes),
    fairly_activemin = sum(FairlyActiveMinutes),
    lightly_activemin = sum(LightlyActiveMinutes),
    sed_min = sum(SedentaryMinutes),
    wk_calories = sum(Calories),
    wk_timebed = sum(TotalTimeInBed),
    wk_asleep = sum(TotalMinutesAsleep)
  )
```

Examining the weekly data reveals a significant amount of **missing** data in variables associated with sleep.

We also observe instances of "0" values in data related to activity, which could indicate potential **measurement errors**. Hence, caution is advised when interpreting results derived from this data.

Exploring Daily Data: Summary Statistics

Take a glimpse into the world of daily data as we delve into summary statistics. This overview will provide valuable insights into the distribution and central tendencies of our daily variables. Join me in this concise exploration of the data's statistical landscape.

```
merged_data %>%
  select(TotalSteps,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes,
         SedentaryMinutes,
         Calories) %>%
  summary()
```

```
##    TotalSteps    VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
##  Min.   :    0   Min.   :  0.00    Min.   :  0.00      Min.   :  0
##  1st Qu.: 3795   1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127
##  Median : 7439   Median :  4.00    Median :  7.00      Median :199
##  Mean   : 7652   Mean   : 21.24    Mean   : 13.63      Mean   :193
##  3rd Qu.:10734   3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264
##  Max.   :36019   Max.   :210.00    Max.   :143.00      Max.   :518
##  SedentaryMinutes    Calories
##  Min.   :   0.0   Min.   :   0
##  1st Qu.: 729.0   1st Qu.:1830
##  Median :1057.0   Median :2140
##  Mean   : 990.4   Mean   :2308
##  3rd Qu.:1229.0   3rd Qu.:2796
##  Max.   :1440.0   Max.   :4900
```

The summary statistics reveal that, on average, individuals in this sample take approximately 7652 steps per day. This stands as the only comparable metric with recommended guidelines, as the intensity thresholds in the data differ from those suggested by World Health Organization or American College of Sports Medicine. Notably, the sample engages in more minutes of sedentary physical activity compared to very active or fairly active minutes, suggesting a need for reduction in sedentary behavior.

Crafting a Subset Dataset for Histograms

Embark on the creation of a tailored dataset, carefully selecting the specific data points intended for histogram plotting. This refined dataset will serve as the canvas for visual representations, offering a focused and insightful perspective on the chosen variables.

```
merged_data_hist <- merged_data %>% select(TotalSteps,
                                           VeryActiveMinutes,
                                           FairlyActiveMinutes,
                                           LightlyActiveMinutes,
                                           SedentaryMinutes,
                                           Calories)
```

Exploring Distributions with Histograms

Dive into a more detailed examination using histograms.

Defining Subplots Function

Let's enhance our visual exploration by creating subplots. Utilizing a function sourced from this link, we'll construct a versatile tool for plotting and analyzing distributions.

```
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
```

```
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

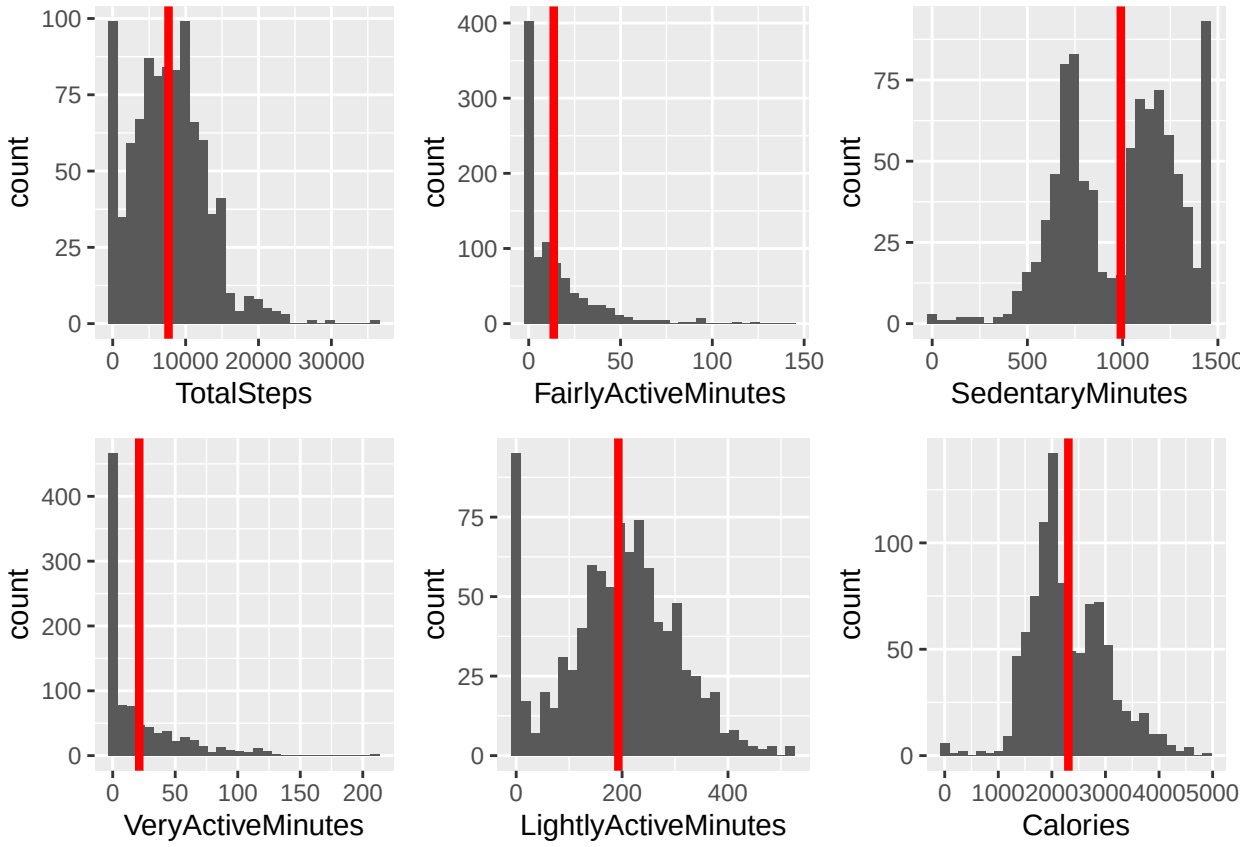
Visualizing Distributions: Variable Histograms

Shift your focus to a visual exploration of individual variable distributions through histograms. This graphical representation will provide a clearer understanding of the data's underlying patterns. Let's unveil the insights hidden within each variable.

```
myplots <- list()  # new empty list
for(i in 1:ncol(merged_data_hist)){
  col <- names(merged_data_hist)[i]
  ggp <- ggplot(merged_data_hist, aes_string(x = col)) +
    geom_histogram(bins = 30) +
    geom_vline(xintercept = mean(merged_data_hist[[col]]), col = "red", lwd=1.5)


  myplots[[i]] <- ggp  # add each plot into plot list
}

multiplot(plotlist = myplots, cols = 3)
```



Analyzing Hourly Steps: Unveiling Patterns

Shift your attention to the hourly steps behavior of each individual, aiming to identify potential patterns within the data. This examination may uncover insights into the temporal distribution of steps throughout the day. Let's explore and decipher the patterns that may emerge from the hourly steps data.

Load dataset

```
steps_hour <- read_csv("data/Fitabase Data 4.12.16-5.12.16/hourlySteps_merged.csv")
```

Converting Time: From Character to Date, Then to Hour Format

To facilitate our analysis, let's undergo a two-step process. First, we'll convert the hour from a character format to a date format. Subsequently, we'll transform it into an hour format. This conversion will streamline our exploration of hourly steps behavior for each individual.

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

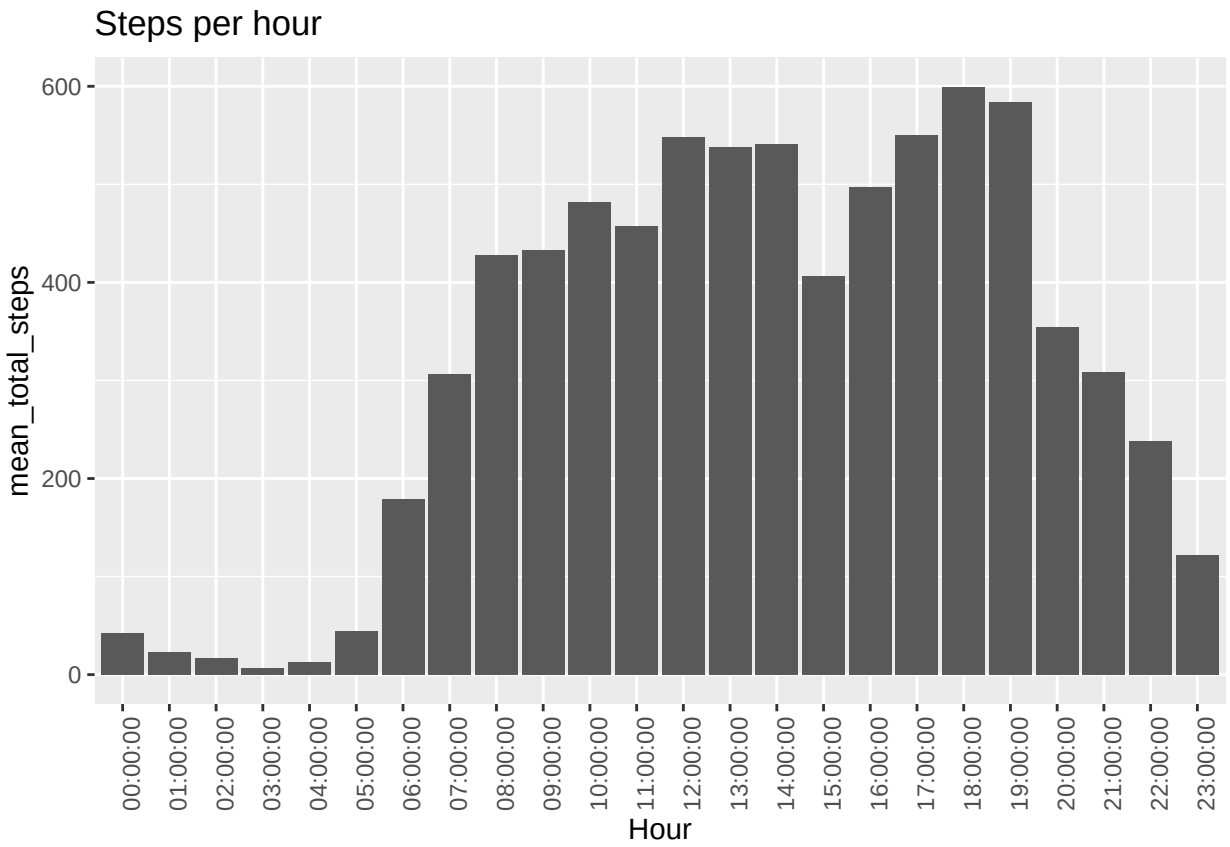
Estimating Hourly Steps

Moving forward, let's proceed with estimating the steps given per hour. This calculation is pivotal in unraveling the hourly distribution of steps for each individual, shedding light on potential patterns and trends within the data.

```
steps_h <- steps_hour %>%
  group_by(Hour) %>%
  drop_na() %>%
  summarise(mean_total_steps = mean(StepTotal))
```

Visualizing Hourly Steps Behavior

```
ggplot(data=steps_h, aes(x=Hour, y=mean_total_steps)) + geom_histogram(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title="Steps per hour")
```



From the depicted plot, it's evident that the average steps per day begin to rise around 6 am, reaching a peak near 7 pm, and then gradually decline toward 9 pm. This observation unveils the daily pattern of steps, showcasing peak activity in the evening hours.

Exploring Weekly Behavior: Network Analysis Preparation

To initiate our exploration of weekly behavior through network analysis, we'll first filter the dataset to include only data from a specified week—in this case, the second week of monitoring. This focused dataset lays the groundwork for uncovering patterns and relationships within the weekly context.

```
data_to_network <- weekly_merged_data %>%
  filter(week == 2)
```

Refining Data for Network Estimation

For accurate estimation of the network, let's further filter the dataset, retaining only the data crucial for the covariance matrix estimation. Notably, we'll exclude the variable related to sleep due to insufficient data for network analysis.

```
data_to_model <- data_to_network %>%
                 ungroup() %>%
                 select(wk_steps,
                        very_activemin,
                         fairly_activemin,
```

```
                        lightly_activemin,
                        sed_min,
                        wk_calories)
```

Renaming variables to a better visualization

In this step I changed the columns names to numbers to allow the construction of a better layout to the graph visualization.

```
colnames(data_to_model)[colnames(data_to_model) == "wk_steps"] ="1"
colnames(data_to_model)[colnames(data_to_model) == "very_activemin"] ="2"
colnames(data_to_model)[colnames(data_to_model) == "fairly_activemin"] ="3"
colnames(data_to_model)[colnames(data_to_model) == "lightly_activemin"] ="4"
colnames(data_to_model)[colnames(data_to_model) == "sed_min"] ="5"
colnames(data_to_model)[colnames(data_to_model) == "wk_calories"] ="6"
```

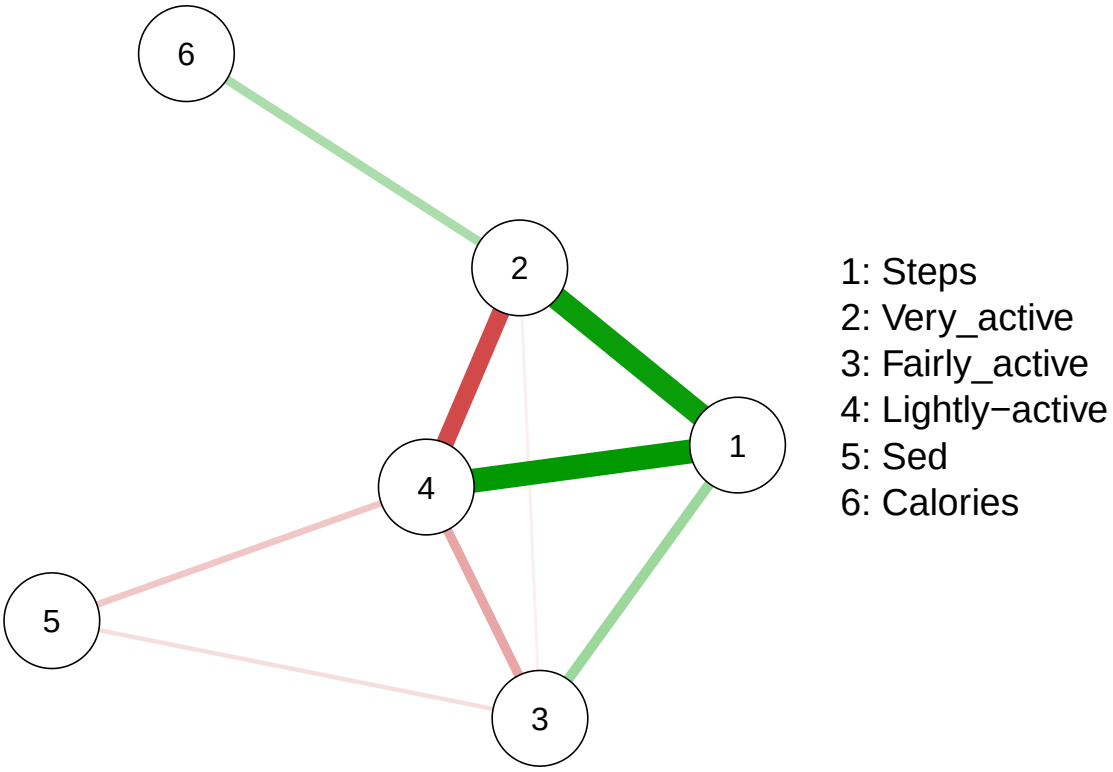
Estimating Covariance Matrix for Network Construction

```
covmat_model <- cov(data_to_model)
```

Visualizing the Network

Shift your attention to the visual representation of our network. This graphical depiction will provide a clear and insightful view of the relationships and connections within the data. Let's explore the network and unravel the patterns and interactions that emerge.

```
cornet_model <- qgraph(covmat_model, layout = "spring", minimum = 0.3, graph = "pcor", nodeNames = c("S
                       edge.labels = FALSE, legend.mode = "names")
```



From the observed network, different **positive** and **negative** relationships emerge among the variables collected from the Fitbit app. In **positive** associations, a notable pattern emerges: individuals who engage in more very active minutes tend to expend a higher number of calories. Additionally, positive correlations manifest between the number of very active minutes and steps, the number of lightly active minutes and steps, as well as between the number of fairly active minutes and steps.

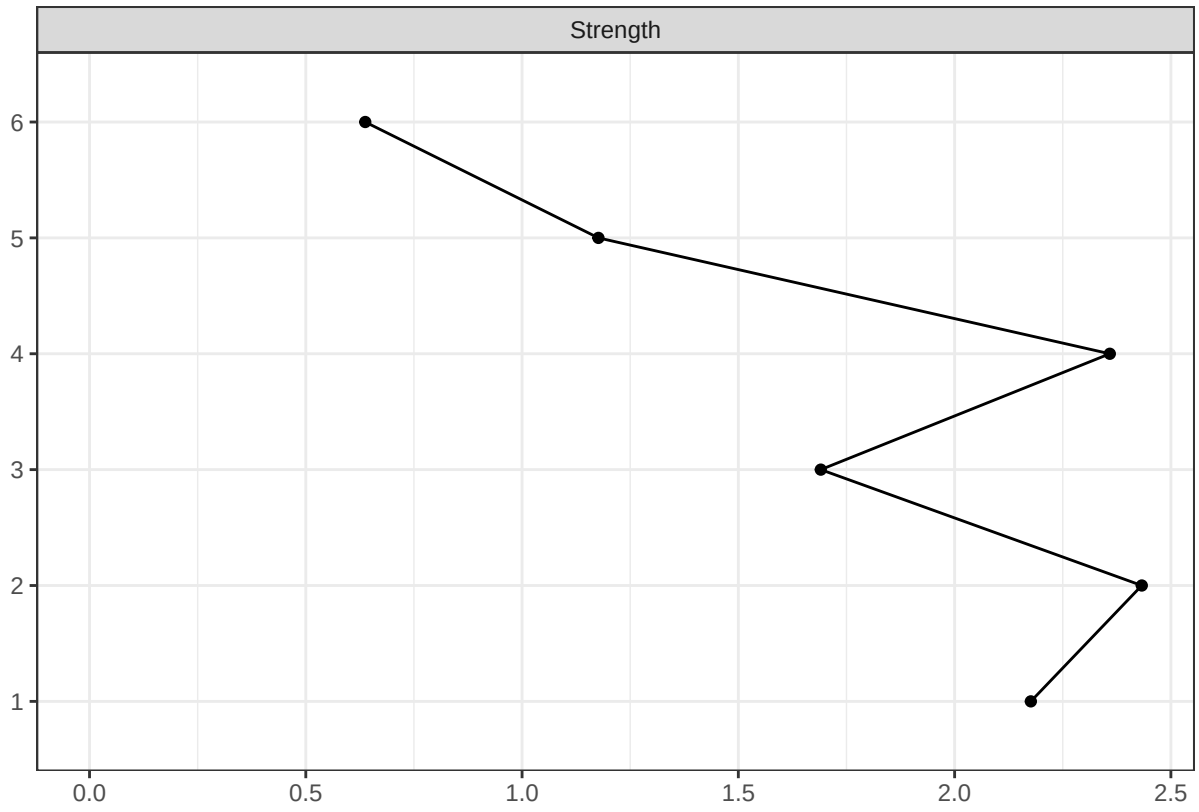
Conversely, the **negative** relationships indicate an inversely proportional association. For example, a decrease in very active minutes is associated with an increase in lightly active minutes, and similarly, a decrease in fairly active minutes is related to an increase in lightly active minutes. Negative correlations are also observed between sedentary minutes and both lightly active and fairly active minutes, as well as between very active minutes and fairly active minutes.

In practical terms, this can be interpreted as follows: for instance, if the goal is to decrease the sedentary time among individuals in this sample, two viable intervention candidates would be increasing lightly active minutes and fairly active minutes. This recommendation stems from the observed negative relationships between these variables and sedentary time.

Examining Key Variables in the Network

Concluding our network exploration, let's direct our attention to identifying the most crucial variables within the network. By assessing importance through the strength centrality metric, we can pinpoint the variables that play a pivotal role in shaping the overall network structure. Let's unveil and analyze the most influential elements in our network.

```
centralityPlot(cornet_model)
```



Derived from the aforementioned results, it is evident that the most crucial variables in the network were lightly active minutes and very active minutes. This implies that, if there is a desire to alter the observed outcomes—specifically, to reduce sedentary time and increase the relative number of steps—these two variables emerge as potential candidates for intervention.

Business Recommendations Summary

Certainly, the data analysis provides a foundation for formulating various recommendations for Bellabeat.

Recommendations for Data Collection Enhancement

1 **Sample Size Expansion**:

- Consider strategies to increase the sample size for a more representative dataset.

2 **Standardized Protocol Implementation**:

- Introduce a standardized data collection protocol to ensure consistency and comparability across all participants.

3 **Uniform Data Collection Periods**:

- Enforce uniformity in data collection periods among individuals to mitigate potential biases.

4 **Comprehensive Sleep and activity intensity Metrics**:

- Refine data collection procedures to encompass a more comprehensive analysis of sleep-related metrics and activity intensity which aligns with that used for the World Health Organization.

5 **Inclusion of Demographic Data**:

- Collect additional demographic information such as sex, age, and socioeconomic status to enrich the dataset and allow for more nuanced analyses.

By addressing these recommendations, Bellabeat can optimize its data collection practices, fortifying the foundation for future data-driven decisions and advancements.

Recommendations for Actionable Insights

1 **Automated Feedback Algorithms**:

- Develop algorithms capable of automatically recognizing and providing feedback during periods of low activity, utilizing insights from hourly data.

2 **Focus on Light and Fairly Active Minutes**:

- Emphasize interventions that aim to increase lightly or fairly active minutes, as these variables play a pivotal role in influencing various outcomes, especially sedentary minutes.

3 **Tailored Weight Management Feedback**:

- Design feedback mechanisms tailored to individuals seeking weight reduction, with a focus on increasing very active minutes to boost calorie expenditure.

4 **Utilize Advanced Data Modeling Tools**:

- Leverage advanced data modeling tools, such as latent class analysis, to identify distinct patterns among individuals, enhancing the precision of interventions.

5 **Individualized Time-Series Analysis**:

- Implement time-series analysis tailored to the individual level to gain insights into the mechanisms underlying data-generated patterns, allowing for the anticipation of periods of inactivity.

6 **Explore Idiographic Networks**:

- Embark on the exploration of idiographic networks to understand how the interaction between variables evolves over time, facilitating the refinement and personalization of interventions.

By incorporating these recommendations, Bellabeat can harness actionable insights from Fitbit data to tailor interventions and enhance the overall user experience and marketing actions.

That concludes my case study. Thank you for taking the time to read it. I welcome any feedback you may have, and appreciate your insights :)