Bella Beat

Vishesh Modi

27/10/2021

Libraries used

So the libraries I’ve used for my case study are tidyverse, lubridate and ggplot

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(dplyr)
library(tibble)
options(scipen = 100)
```

Dataframes

The data frames used to make the analysis has information about dailyActivites, sleepDay merged and weightLogInfo_merged.

```
setwd("D:/Data Analysis/Case Study - bellaBeat/Fitabase Data 4.12.16-5.12.16")
df1 <- read_csv("dailyActivity_merged.csv")
```

```
## Rows: 940 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df2 <- read_csv("sleepDay_merged.csv")
```

```
## Rows: 413 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDay
## dbl (5): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed, TotalHou...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df3 <- read.csv("weightLogInfo_merged.csv")
```

Some Basic Info about the data frame 1

```
colnames(df1)
```

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

```
head(df1) %>% select(Id, ActivityDate, Calories)
```

```
## # A tibble: 6 x 3
##           Id ActivityDate Calories
##        <dbl> <chr>           <dbl>
## 1 1503960366 4/12/2016        1985
## 2 1503960366 4/13/2016        1797
## 3 1503960366 4/14/2016        1776
## 4 1503960366 4/15/2016        1745
## 5 1503960366 4/16/2016        1863
## 6 1503960366 4/17/2016        1728
```

Some Basic Info about the data frame 2

```
colnames(df2)
```

```
## [1] "Id"                 "SleepDay"           "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"     "TotalHoursAsleep"
```

```
head(df2) %>% select(Id, SleepDay, TotalHoursAsleep)
```

```
## # A tibble: 6 x 3
##           Id SleepDay              TotalHoursAsleep
##        <dbl> <chr>                            <dbl>
## 1 1503960366 04-12-2016 00:00                  5.45
## 2 1503960366 4/13/2016 12:00:00 AM             6.4
## 3 1503960366 4/15/2016 12:00:00 AM             6.87
## 4 1503960366 4/16/2016 12:00:00 AM             5.67
## 5 1503960366 4/17/2016 12:00:00 AM            11.7
## 6 1503960366 4/19/2016 12:00:00 AM             5.07
```

Some Basic Info about the data frame 3

```
colnames(df3)
```

```
## [1] "Id"             "Date"           "WeightKg"       "WeightPounds"
## [5] "Fat"            "BMI"            "IsManualReport" "LogId"
```

```
head(df3) %>% select(Id, Date,WeightKg)
```

```
##           Id                  Date WeightKg
## 1 1503960366  5/2/2016 11:59:59 PM     52.6
## 2 1503960366  5/3/2016 11:59:59 PM     52.6
## 3 1927972279  4/13/2016 1:08:52 AM    133.5
## 4 2873212765 4/21/2016 11:59:59 PM     56.7
## 5 2873212765 5/12/2016 11:59:59 PM     57.3
## 6 4319703577 4/17/2016 11:59:59 PM     72.4
```

Now renaming some column names

```
df1 <- plyr :: rename(df1, c("ActivityDate" = "Date"))
df2 <- plyr :: rename(df2, c("SleepDay" = "Date"))
```

Drop Unecessary Columns

```
df1 <- select(df1, Id, Date, Calories)
df2 <- select(df2, Id, Date, TotalHoursAsleep)
df3 <- select(df3, Id, Date, WeightKg)
colnames(df1)
```

```
## [1] "Id"       "Date"     "Calories"
```

```
colnames(df2)
```

```
## [1] "Id"               "Date"             "TotalHoursAsleep"
```

```
colnames(df3)
```

```
## [1] "Id"       "Date"     "WeightKg"
```

Check for number of rows and unique data

```
n_distinct(df1$Id)
```

```
## [1] 33
```

```
n_distinct(df2$Id)
```

```
## [1] 24
```

```
n_distinct(df3$Id)
```

```
## [1] 8
```

Now merging the dataset will help to elimate the extra data

```
df4 <- merge(df2, df1, by = "Id")
df5 <- merge(df4, df3, by = "Id")
head(df5)
```

```
##           Id           Date.x TotalHoursAsleep    Date.y Calories
## 1 1503960366 05-01-2016 00:00         6.150000 4/24/2016     1788
## 2 1503960366 05-01-2016 00:00         6.150000 4/24/2016     1788
## 3 1503960366 05-02-2016 00:00         4.616667 4/24/2016     1788
## 4 1503960366 05-02-2016 00:00         4.616667 4/24/2016     1788
## 5 1503960366 05-01-2016 00:00         6.150000 4/21/2016     1775
## 6 1503960366 05-01-2016 00:00         6.150000 4/21/2016     1775
##                   Date WeightKg
## 1 5/3/2016 11:59:59 PM     52.6
## 2 5/2/2016 11:59:59 PM     52.6
## 3 5/3/2016 11:59:59 PM     52.6
## 4 5/2/2016 11:59:59 PM     52.6
## 5 5/3/2016 11:59:59 PM     52.6
## 6 5/2/2016 11:59:59 PM     52.6
```

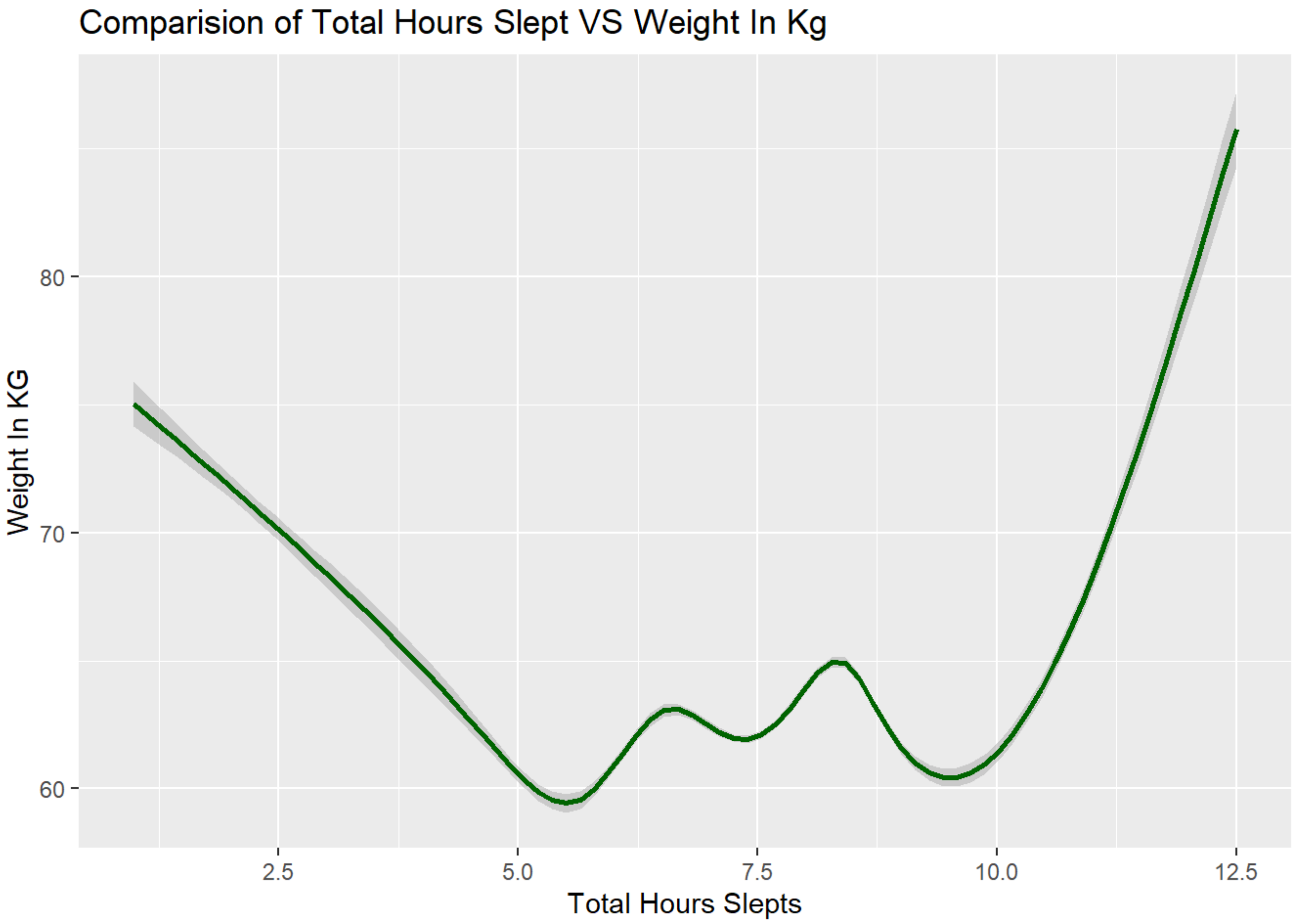
Some conversions into categories

```
df5 <- df5 %>%
  mutate(sleepCategories = case_when(
    TotalHoursAsleep >6 & TotalHoursAsleep <= 8 ~ "6h-8h",
    TotalHoursAsleep > 8 ~ "> 8h",
    TRUE ~ "< 6h"
  ))
```

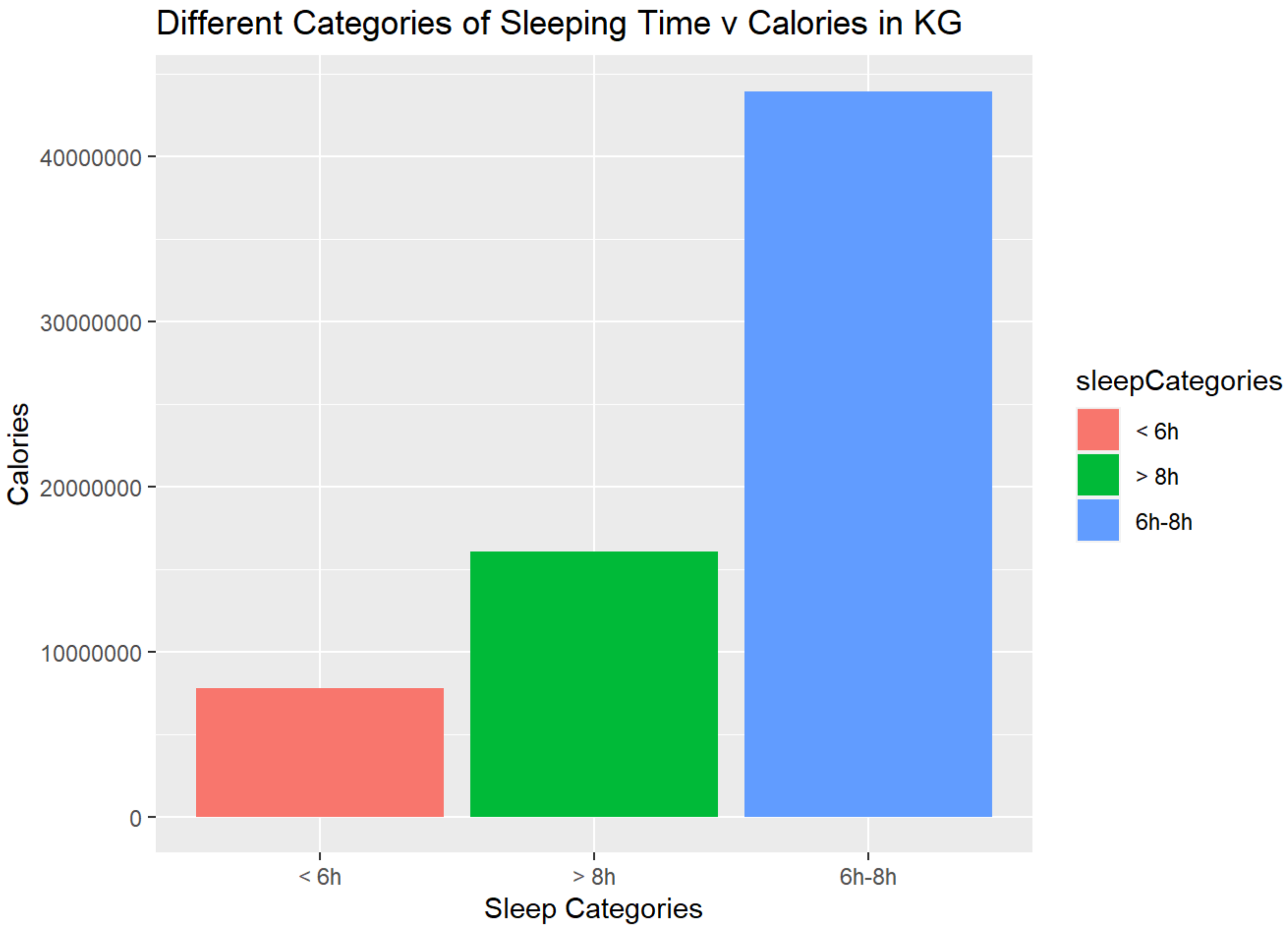
Some Data Visualizations

```
ggplot(data = df5) + geom_smooth(mapping = aes(x = TotalHoursAsleep, y = WeightKg), color =
'darkgreen') + labs(title = "Comparision of Total Hours Slept VS Weight In Kg", x = "Total Ho
urs Slepts", y = "Weight In KG")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
ggplot(data = df5) + geom_col(mapping = aes(x = sleepCategories, y = Calories, fill = sleepCa
tegories)) + labs(title = "Different Categories of Sleeping Time v Calories in KG", x = "Slee
p Categories", y = "Calories")
```



Summary

1. There are two graphs showing relationships of Total Hours Slept v Weight In KG and Sleep categories v Calories burnt.
2. The first one shows that the person who sleeps between 6 - 8hrs is healthier when compared to person who sleeps less than 6 hrs and more than 8hrs
3. The second one shows that the person who sleeps between between 6 - 8hrs lose burns more calories when compared to person who sleeps less than 6 hrs and more than 8hrs.

Recommendations

1. So there’s a clear relationship between weight and sleep and calories and sleep, so the app should focus more on giving tips and motivate it’s users that why a healthy sleep is required for human body.
2. The app should also feature functions to track sleeping activities like when is the best time to sleep, provide exclusive alarm features to wake up early.