Fitbit_Part1

Litta Rizki A

3/8/2022

This markdown was part of how I used R to do my research. The rationale for including parts in each markdown was so that I could keep track of my R learning, which includes:

- how I rewrote the code
- how I dive further into the data to find answers to my questions about it

Let's get this party started, shall we?

IMPORTING PACKAGES & DATA

Packages

```
library(skimr)
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     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
```

Data

```
dailydata <- 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.
```

```
head(dailydata)
```

```
## # 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>
```

EDA

It's better to **know your data** before we start manipulating or plotting the data. which consists of:

- What data type is it?
- How many columns and rows?

I'm going to use the str() function for this.

```
str(dailydata)
```

```
## spec_tbl_df [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                       : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate             : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps               : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance            : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes         : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                 : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
```

```
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

As you can see, we've already identified our first issue, which is **the data type**. It was classified as "num" in the Id column, for example. **If we allow this to happen, it will be difficult to manipulate data** based on the Id.

Second, instead of being classified as "Date," the ActivityDate is now classified as "character," which will lead to data bias in the future if we allow both of those to slide.

So, let's convert the data type to a more proper.

```
dailydata$ActivityDate <- mdy(dailydata$ActivityDate)

dailydata$Id <- as.character(dailydata$Id)

str(dailydata)
```

```
## spec_tbl_df [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                       : chr [1:940] "1503960366" "1503960366" "1503960366" "1503960366" ...
##  $ ActivityDate             : Date[1:940], format: "2016-04-12" "2016-04-13" ...
##  $ TotalSteps               : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance            : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes         : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                 : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
```

```
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

CREATING SUBSET

I believe we have already addressed what needs to be adjusted at this time. If we want to get a deeper analysis into all of the data, it will be much ahead of for a rookie data analyst like me. That is why it is critical to **begin with the simplest scenario first, followed by an update on what I should have done differently with the data**.

As a result, after I converted the data type to the proper one. I'd like to starting the data manipulation process by **producing a subset of the main dataset**.

I began by grouping (aggregating) each piece of data based on their Id and the date.

Perday Subset

```
PerDay <- dailydata %>% count(ActivityDate, sort = T)

names(PerDay)[2] <- 'User_perday'

PerDay <- dailydata %>% group_by(ActivityDate) %>%
  summarise(total_dailySteps = sum(TotalSteps)
            ,total_dailydistance = sum(TotalDistance)
            ,total_dailycalories = sum(Calories))

PerDay
```

```
## # A tibble: 31 x 4
##    ActivityDate total_dailySteps total_dailydistance total_dailycalories
##    <date>                  <dbl>                <dbl>                <dbl>
##  1 2016-04-12             271816                197.                78893
##  2 2016-04-13             237558                168.                75459
##  3 2016-04-14             255538                185.                77761
##  4 2016-04-15             248617                174.                77721
##  5 2016-04-16             277733                201.                76574
##  6 2016-04-17             205096                145.                71391
##  7 2016-04-18             252703                181.                74668
##  8 2016-04-19             257557                188.                75491
##  9 2016-04-20             261215                190.                76647
## 10 2016-04-21             263795                193.                77500
## # ... with 21 more rows
```

PerUser Subset

```
PerUser <- dailydata %>% group_by(Id) %>%
  summarise(UserSteps = sum(TotalSteps)
            ,UserDistance = sum(TotalDistance)
            ,UserCalories = sum(Calories))

PerUser
```

```
## # A tibble: 33 x 4
```

```
##    Id         UserSteps UserDistance UserCalories
##    <chr>          <dbl>        <dbl>        <dbl>
##  1 1503960366    375619        242.         56309
##  2 1624580081    178061        121.         45984
##  3 1644430081    218489        159.         84339
##  4 1844505072     79982         52.9        48778
##  5 1927972279     28400         19.7        67357
##  6 2022484408    352490        251.         77809
##  7 2026352035    172573        107.         47760
##  8 2320127002    146223         98.8        53449
##  9 2347167796    171354        114.         36782
## 10 2873212765    234229        158.         59426
## # ... with 23 more rows
```

As you can see, I'm simply using three variables at the moment to get a basic overview of the data.

As I was producing this subset, I began to consider how this subject usage time, and the data had already provided me with the information I required.

That's the exciting part about working as a data analyst. Yes, we may begin with some underlying or narrow considerations, but as you dig further into the data, plenty of new problems will arise. It's like a never-ending activity, and there'll always be something fresh if we can put our ideas into action by using the practical capabilities required to realize them.

Enough chit-chat, let's get begin on creating a new subset for part 1 markdown.

Usage Subset

```
MinutesUsage <- dailydata %>%
  select(Id, ActivityDate, VeryActiveMinutes,FairlyActiveMinutes, LightlyActiveMinutes)

head(MinutesUsage)
```

```
## # A tibble: 6 x 5
##   Id         ActivityDate VeryActiveMinutes FairlyActiveMinutes LightlyActiveMi~
##   <chr>      <date>                  <dbl>               <dbl>            <dbl>
## 1 1503960366 2016-04-12                 25                  13              328
## 2 1503960366 2016-04-13                 21                  19              217
## 3 1503960366 2016-04-14                 30                  11              181
## 4 1503960366 2016-04-15                 29                  34              209
## 5 1503960366 2016-04-16                 36                  10              221
## 6 1503960366 2016-04-17                 38                  20              164
```

Based on Date

```
DailyUsage <- MinutesUsage %>% group_by(Id) %>%
  summarise(VeryActive = sum(VeryActiveMinutes),FairlyActive = sum(FairlyActiveMinutes)
            ,LightlyActive = sum(LightlyActiveMinutes))

head(DailyUsage)
```

```
## # A tibble: 6 x 4
##   Id         VeryActive FairlyActive LightlyActive
##   <chr>           <dbl>        <dbl>         <dbl>
## 1 1503960366       1200          594          6818
## 2 1624580081        269          180          4758
```

```
## 3 1644430081        287          641          5354
## 4 1844505072          4           40          3579
## 5 1927972279         41           24          1196
## 6 2022484408       1125          600          7981
```

Based on Id

```
UsersUsage <- MinutesUsage %>% group_by(ActivityDate) %>%
  summarise(VeryActive = sum(VeryActiveMinutes),FairlyActive = sum(FairlyActiveMinutes)
            ,LightlyActive = sum(LightlyActiveMinutes))

head(UsersUsage)
```

```
## # A tibble: 6 x 4
##   ActivityDate VeryActive FairlyActive LightlyActive
##   <date>            <dbl>        <dbl>         <dbl>
## 1 2016-04-12          736          259          6567
## 2 2016-04-13          671          349          5998
## 3 2016-04-14          691          409          6633
## 4 2016-04-15          633          326          7057
## 5 2016-04-16          891          484          6202
## 6 2016-04-17          605          379          5291
```

I believe that concludes Part 1 of this series. We've already created a few necessary subsets, and we'll start plotting in the next section.

for the original data you can access this link or you can just catch me up to my e-mail

It would be fantastic if you could provide me with some feedback.

Thank you, and I hope to see you again soon!!

To Be Continue