Analysing activity monitoring Data

L

2 12 2021

Activity Monitoring Data of an anonymous human being

In this file, the process of analysing activity monitoring data of an human will be loaded and analysed. Below, each step will be explained and supported with an R-Code.

Loading required R packages

First, loading R packages is required.

```
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   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

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

```
library(lubridate)
```

```
##
## Attache Paket: 'lubridate'
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     date, intersect, setdiff, union
```

Loading and preprocessing the data

Load the data by the following code:

```
filename <- "repdata_data_activity.zip"
if (!file.exists(filename)){
  download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", filename)
  unzip(filename)
  }
dir <- dir()
importfile <- dir[1]
monitordata <- read.table(importfile, header = T, sep="," , na.strings = "NA")
```

check the structure:

```
str(monitordata)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Adjust class of the variable date

```
monitordata$date <- as.Date(monitordata$date, "%Y-%m-%d")
```

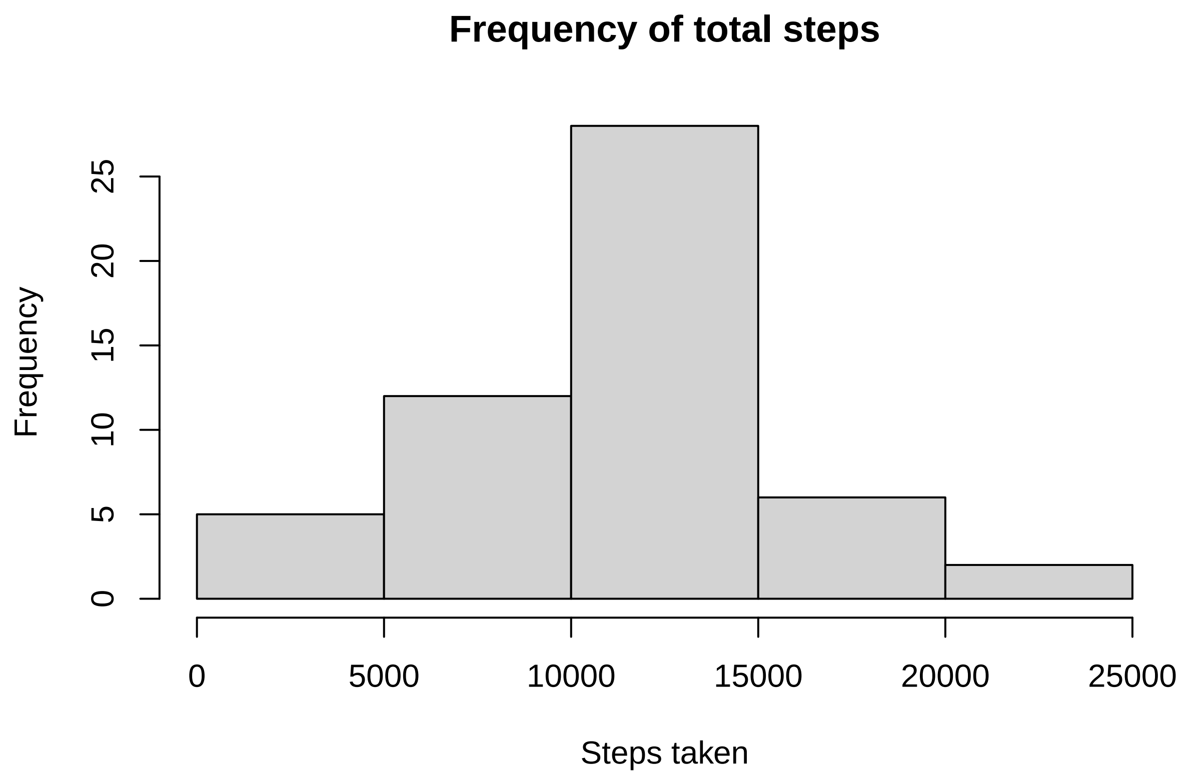
What is mean total number of steps taken per day?

Calculate the total number of steps taken per day by the following:

```
total <- monitordata %>% group_by(date) %>% summarize(total_steps = sum(steps, na.rm = F))
```

Drawing a histogram to show how often certain levels of steps are achieved during the observation period

```
hist(total$total_steps, xlab = "Steps taken", ylab = "Frequency", main = "Frequency of total steps")
```



Summarizing the mean and median of steps during the observation period

```
summary <- total %>% summarize(mean = mean(total_steps, na.rm=T), median= median(total_steps, na.rm= T)
summary
```

```
## # A tibble: 1 x 2
##      mean median
##     <dbl>  <int>
## 1 10766.  10765
```

The mean is 10766.19 and the median is 10765.

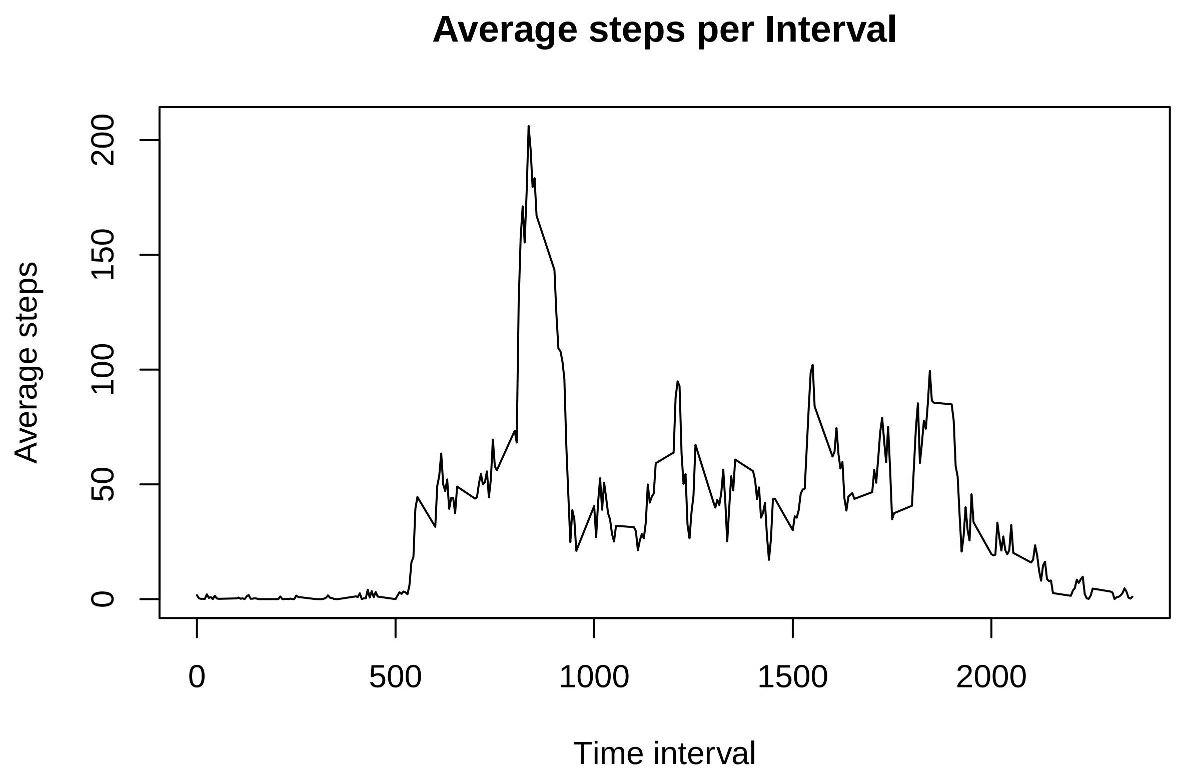
What is the average daily activity pattern?

First, it's necessary to calculate the mean per interval.

```
intervaldata <- monitordata %>% group_by(interval) %>% summarize (mean_interval = mean(steps, na.rm=T))
```

Then it's possible to draw a time series plot, having the intervals on the x-axis and the average number of steps per interval on the y-axis.

```
with(intervaldata, plot(interval, mean_interval, xlab = "Time interval", ylab= "Average steps", type =
```



To investigate, in which interval the subject was most active during the observation period, check it by:

```
maxactivity <- intervaldata %>% filter(mean_interval == max(mean_interval))
```

The most active interval is 835 performing on average 206.17.

Imputing missing values

This data set contains missing values. Investigate how many by

```
nNA <- nrow(monitordata[is.na(monitordata$steps),])
nNA
```

```
## [1] 2304
```

There are 2304 missing values. It is possible to impute these missing values as missing values max introduce bias to analysis. Certain approaches exits, for example imputing with the mean/ median per day or with the mean per interval. In this examination, the latter one was choosen. First, save the data set only containing missing observations (i.e. all rows having the value NA in the column 'steps'). Then extract a variable `missingintervals` listing all intervals effected. Next, the mean steps per interval derived from the previous question serves to fill in the missing values, filtered on the `missingintervals` and saved as a new dataset. This new dataset then is to be merged with the dataset containing only the missing data and finally, rename the variable mean to steps.

```
missings <-  monitordata[is.na(monitordata$steps),]

missingintervals <- unique(missings$interval)

imputingdata <- intervaldata %>% filter(interval %in% missingintervals) %>% select(interval, mean_int

#merge datasets to fill in NAs
imputedmissings <- merge(missings, imputingdata, by ="interval")

# rename column
imputedmissings <- imputedmissings %>% select(-steps) %>% rename(steps = mean_interval) %>% select(st
```

Then, all rows previously having missing values in the column steps are now imputed by the mean of that certain interval. This dataset finally can be row-wise binded to the original dataset resulting in complete records over the whole observation period.

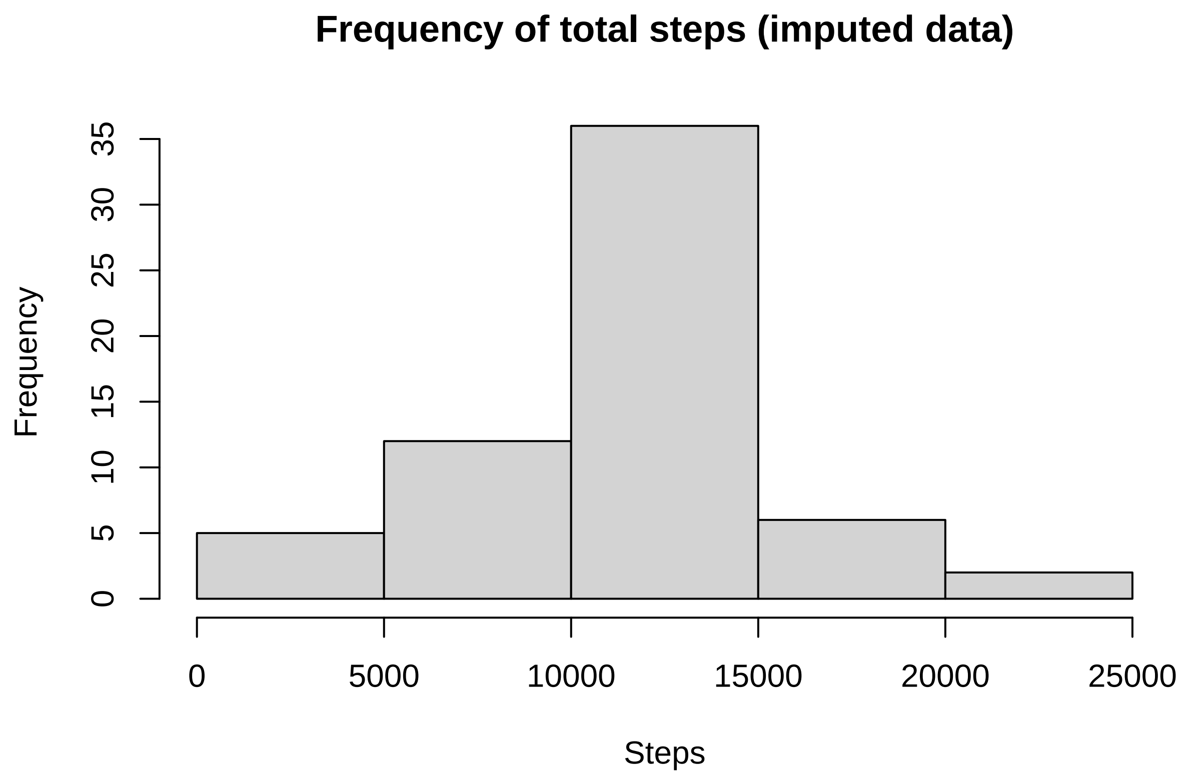
```
imputeddata <- rbind(monitordata[!is.na(monitordata$steps),], imputedmissings)
imputeddata <- imputeddata %>% arrange(date, interval)
```

Let's repeat the steps done in previous exploratoy analysis, first by calculating all steps performed wihtin a day.

```
imputed_totalsteps <- imputeddata %>% group_by(date) %>% summarize(totalsteps = sum(steps))
```

Now, we can visually examine if the count of total steps during the observation period has changed by drawing a histogram.

```
with(imputed_totalsteps, hist(totalsteps, xlab= "Steps", ylab="Frequency", main = "Frequency of total s
```



With having complete records, now lets see if the mean and median has changed in comparison to the datset with missing values. Firstly, calculating mean and median.

```
summary_imputeddata <- imputed_totalsteps %>% summarize(imputed_mean = mean(totalsteps), imputed_median
summary_imputeddata
```

```
## # A tibble: 1 x 2
##   imputed_mean imputed_median
##          <dbl>          <dbl>
## 1       10766.         10766.
```

Now its possible to check out the differnece betwen the means and medians.

```
difference_mean <- summary_imputeddata[1] - summary[1]
difference_mean
```

```
##   imputed_mean
## 1            0
```

```
difference_median <- summary_imputeddata[2] - summary[2]
difference_median
```

```
##   imputed_median
## 1       1.188679
```

The difference in the mean between the complete and missing records is `difference_mean`. There is literally no change after having imputed the missing values. The difference in the median however, has slightly increased by `difference_median` closing the gap to the mean. Summarizing for these records, the imputation effected in a way that mean = median. Imagine the distribution on a curve, the imputation leads to perfect symmetry.

Are there differences in activity patterns between weekdays and weekends?

Activity levels may differ between working days and weekends. Let's explore how this subject behaves, is this person more active during the week or during weekends? To answer this, we need to insert a new variable to the complete dataset called "weekdays". The version of this R studio was German, so a translation is made into English. When having the weekdays, its possible to group according to weekdays (monday to friday) and to weekends (saturdays and sundays)

```
#creating new variable (translate to English)
imputeddata <- imputeddata %>% mutate(weekday = weekdays(date))
imputeddata$weekday <-
  case_when(imputeddata$weekday == "Montag" ~ "monday",
            imputeddata$weekday == "Dienstag" ~ "tuesday",
            imputeddata$weekday == "Mittwoch" ~ "wednesday",
            imputeddata$weekday == "Donnerstag" ~ "thursday",
            imputeddata$weekday == "Freitag" ~ "friday",
            imputeddata$weekday == "Samstag" ~ "saturday",
            imputeddata$weekday == "Sonntag" ~ "sunday",
  )
#define if weekday or weekend
imputeddata <-
  imputeddata %>% mutate(weekend = case_when(
    weekday == "saturday" | weekday == "sunday" ~ "yes",
    TRUE ~ "no"
  ))
```

Let's prepare for visually representation between the two groups. Firstly calculate the mean of steps during the intervals for each, weekdays and weekends.

```
intervalsweekends <- imputeddata %>% group_by(weekend, interval) %>% summarize(mean = mean(steps))
```

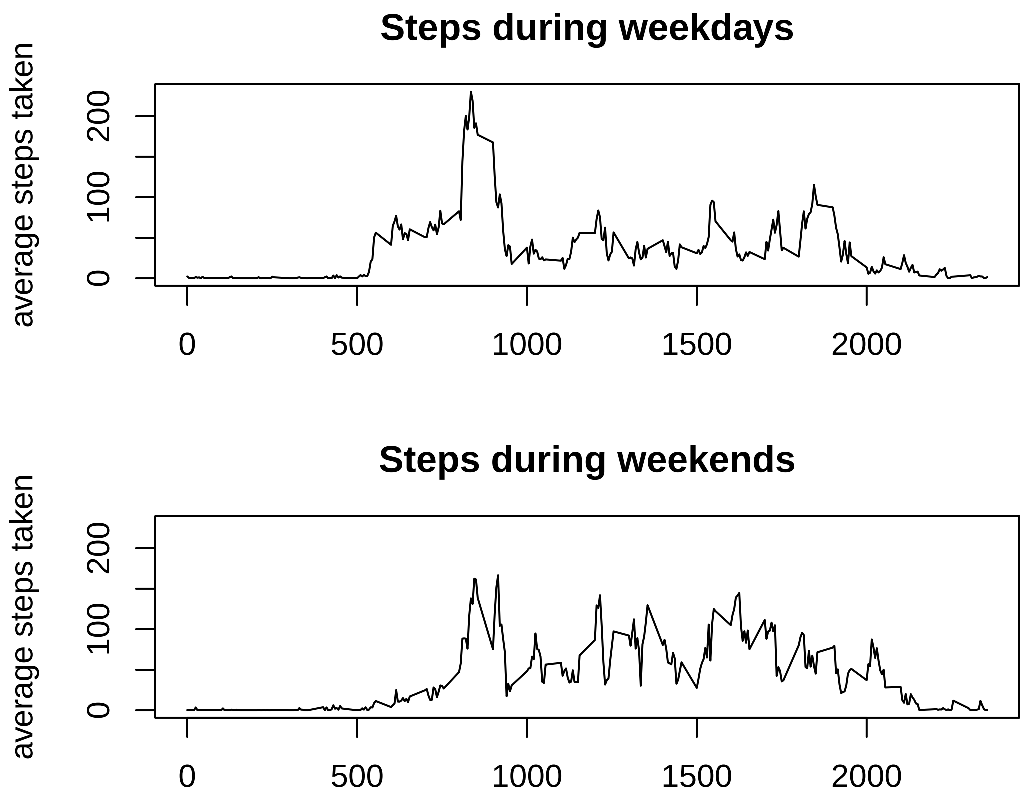
```
## `summarise()` has grouped output by 'weekend'. You can override using the `.groups` argument.
```

```
intervalsweekends
```

```
## # A tibble: 576 x 3
## # Groups:   weekend [2]
##    weekend interval   mean
##    <chr>      <int>  <dbl>
##  1 no             0 2.25
##  2 no             5 0.445
##  3 no            10 0.173
##  4 no            15 0.198
##  5 no            20 0.0990
##  6 no            25 1.59
##  7 no            30 0.693
##   8 no             35 1.14
##   9 no             40 0
## 10 no             45 1.80
## # ... with 566 more rows
```

Now we can code a panel plot showing time series and thus representing the activity throughout the day. To make sure both plots have the same y-axis range, a variable is created so that the axis limits do not disturb the impression.

```
  #to have the same y-axis limits
  rng <- range(intervalsweekends$mean)
par(mfrow = c(2,1), mar= c(3,5,3,5))
with(intervalsweekends[intervalsweekends$weekend == "no",], plot(interval, mean, xlab= "Interval", ylab=
with(intervalsweekends[intervalsweekends$weekend == "yes",], plot(interval, mean, xlab= "Interval", ylal
```



This subject is more active during the weekdays, however there is a clear peek in the morning. During the rest of the day the levels fluctuate slightly. In weekends, the subject doesn't have a peek and is takes more steps throughout the day.