Reproducible Research

Paola Pileri

May 18, 2020

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit (http://www.fitbit.com), Nike Fuelband (http://www.nike.com/us/en_us/c/nikeplusfuelband), or Jawbone Up (https://jawbone.com/up). These type of devices are part of the "quantified self" movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain underutilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data. This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

Loading and preprocessing the data

```
rm(list=ls())
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
data<-read.csv("activity.csv", sep=',', header=TRUE)
names(data)
```

```
## [1] "steps"    "date"     "interval"
```

```
dim(data)
```

```
## [1] 17568     3
```

```
str(data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
head(data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

Processing the data for analysis

```
data <- mutate(data, hour = interval %/% 100, minute = interval %% 100)
```

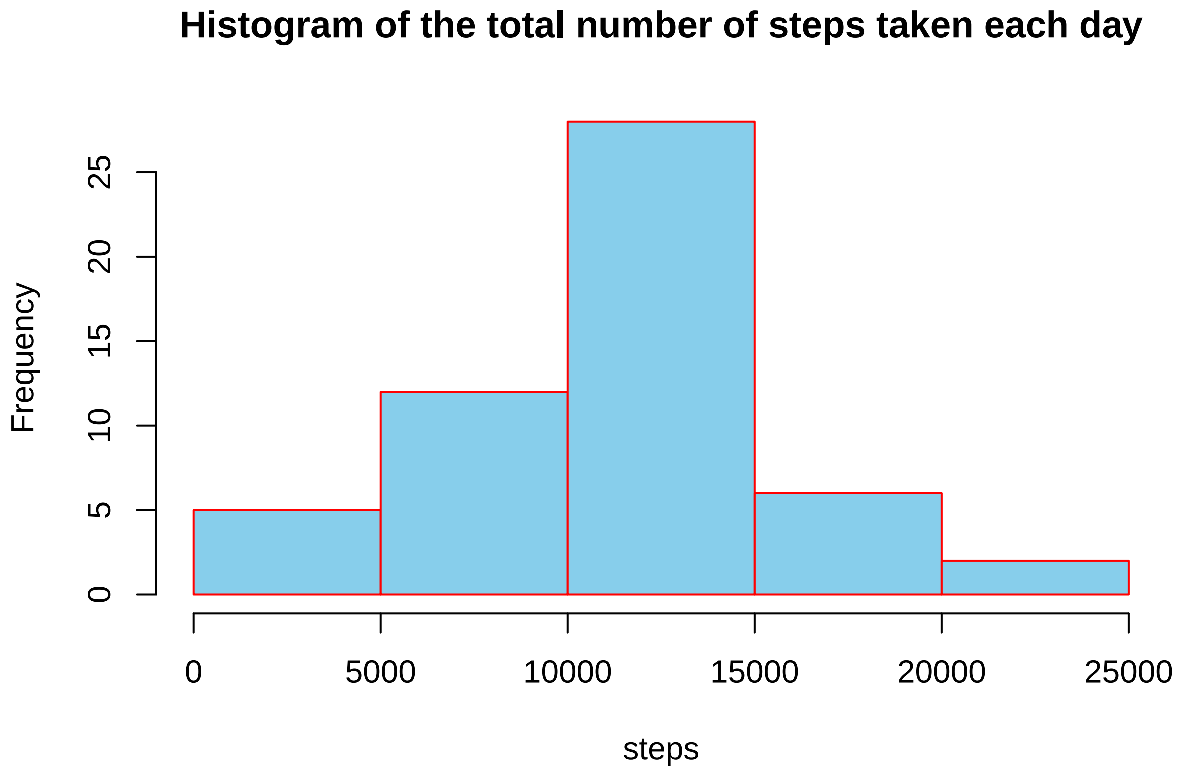
What is mean total number of steps taken per day?

Calculating total number of steps per day

```
daily<-c()
for (i in 1:61){
    start<-(i-1)*288+1
    last<-(i-1)*288+288
    temp<-data[start:last,1]
    daily<-c(daily,sum(temp))
}
```

Making a histogram of the total number of steps taken each day

```
daily_noNA<-daily[!is.na(daily)]
hist(daily_noNA, xlab="steps",ylab="Frequency",col="skyblue",border="red", main="Histogram of the total
```



Calculating the mean and median of the total number of steps taken per day.

```
mean(daily,na.rm=T)
```

```
## [1] 10766.19
```

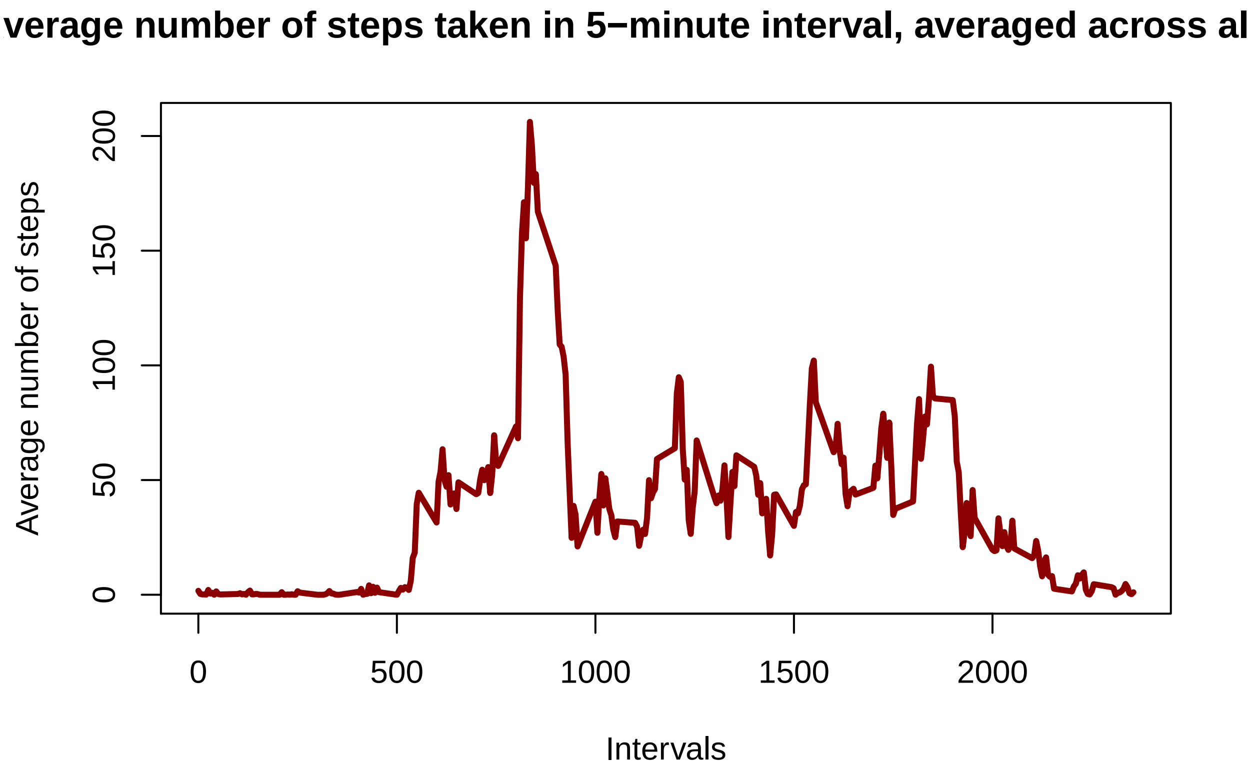
```
median(daily,na.rm=T)
```

```
## [1] 10765
```

What is the average daily activity pattern?

Making a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all day.

```
x<-data[,1]
y<-matrix(x,288,61)
five_average<-apply(y,1,mean,na.rm=TRUE)
plot(data$interval[1:288],five_average, type='l',col='darkred', xlab='Intervals',lwd=3, ylab='Average nu
```



Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
hr<-data$hour[1:288]
min<-data$minute[1:288]
hr_max<-hr[which(five_average==max(five_average))]
min_max<-min[which(five_average==max(five_average))]

cat('The maximum number of steps occurs at',hr_max,':',min_max,'AM')
```

```
## The maximum number of steps occurs at 8 : 35 AM
```

Imputing missing values

Calculating total number of missing values

```
sum(is.na(data[,1]))
```

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

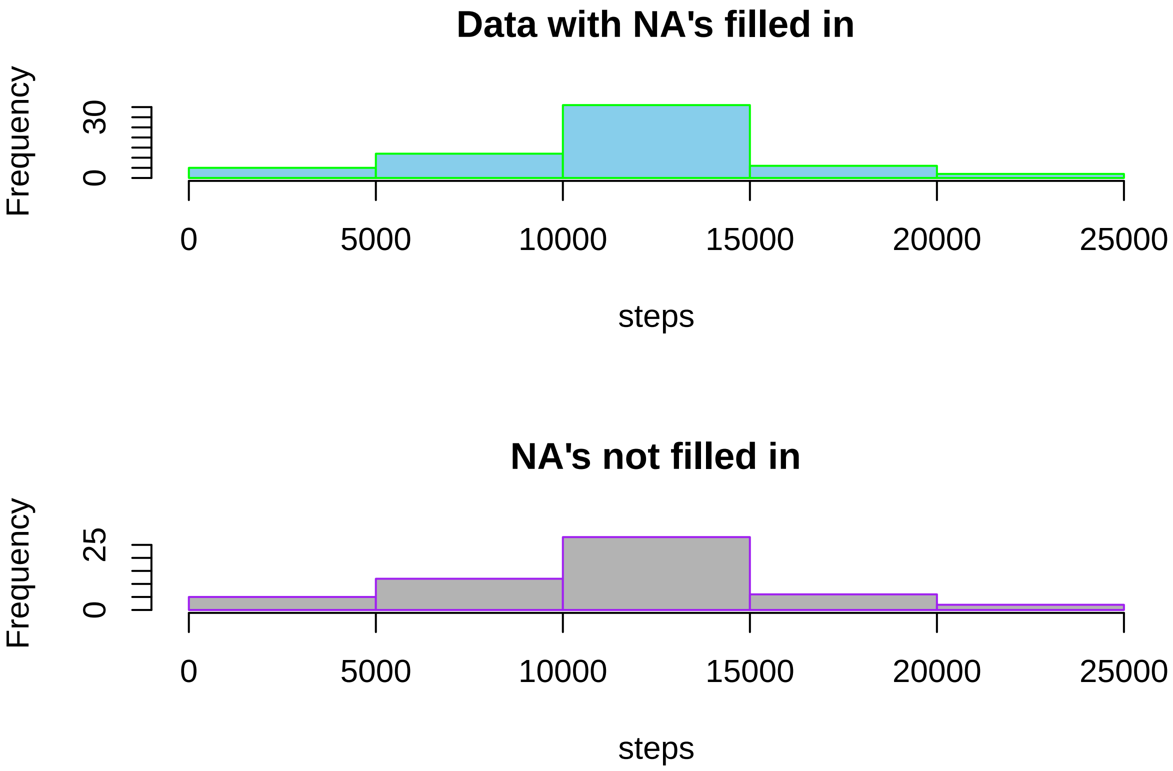
Filling in all of the missing values in the dataset

```
five_average_rep<- rep(five_average,61)
data1<-data
for (i in 1:length(data1[,1])){
    if(is.na(data1[i,1])==TRUE){
        data1[i,1]= five_average_rep[i]
    }}
```

Making a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day.

```
daily1<-c()
for (i in 1:61){
    start<-(i-1)*288+1
    last<-(i-1)*288+288
    temp<-data1[start:last,1]
    daily1<-c(daily1,sum(temp))
}

par(mfrow=c(2,1))
hist(daily1, xlab="steps",ylab="Frequency", main="Data with NA's filled in",border='green',col="skyblue
hist(daily_noNA, xlab="steps",ylab="Frequency", main="NA's not filled in",border='purple',col="gray70",
```



Calculating mean and median of total number of steps daily

```
mean(daily1)
```

```
## [1] 10766.19
```

```
median(daily1)
```

```
## [1] 10766.19
```

Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

Yes, they show diferreneces in the median and in the histograms. imputing missing data on the estimates of the total daily number of steps changes the median, and the distribution as as can be seen from the histograms.Based on the method used for filling in missing values, we can get different mean and median values. The histogram can also be different based on the strategy we used to fill in the missing values.

Are there differences in activity patterns between weekdays and weekends?

Creating a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

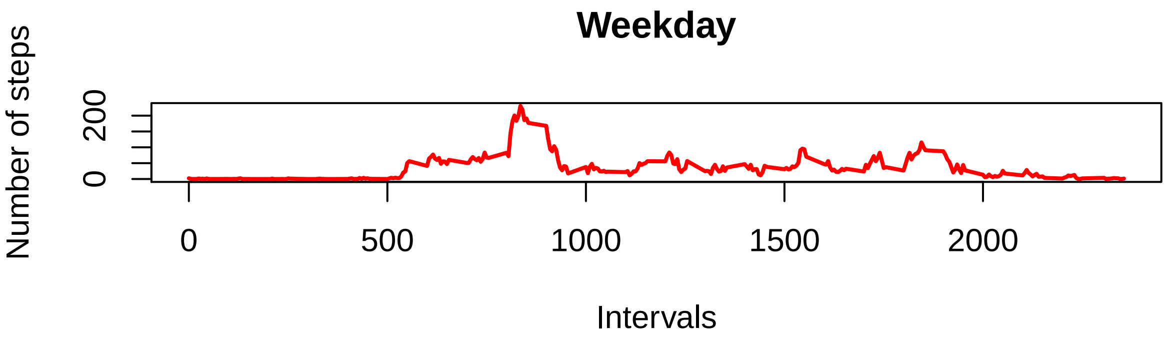
```
data1$date<-as.Date(data1$date)
data1$day<-weekdays(data1$date)
data1_weekdays<-data1[(!data1$day %in% c("Saturday","Sunday")),]
data1_weekend<-data1[(data1$day %in% c("Saturday","Sunday")),]
weekday_steps<-data1_weekdays[,1]
temp<-matrix(weekday_steps,nrow=288)
weekday_steps_average<-apply(temp,1,mean)
weekend_steps<-data1_weekend[,1]
temp<-matrix(weekend_steps,nrow=288)
weekend_steps_average<-apply(temp,1,mean)
```

Making a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

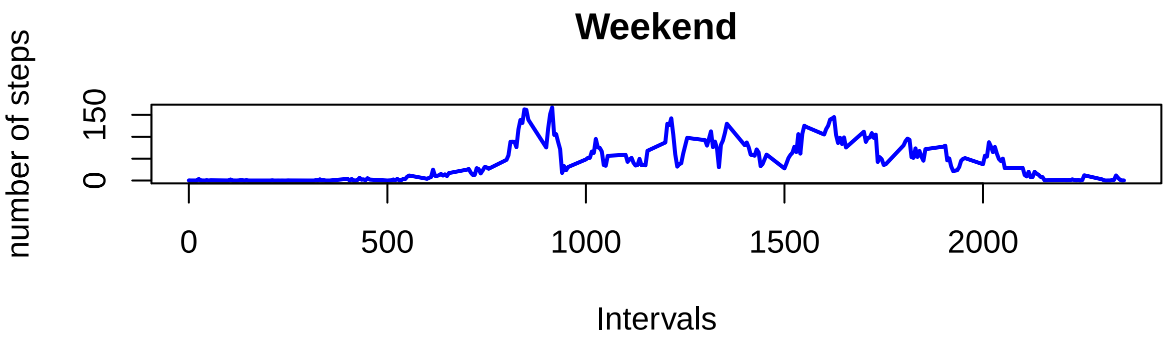
```
par(mfrow=c(2,1))

plot(data$interval[1:288],weekday_steps_average, type="l",xlab='Intervals',ylab="Number of steps",
     col='red',lwd=2, main="Weekday")

plot(data$interval[1:288],weekend_steps_average, type="l", xlab='Intervals',ylab="number of steps",
     col='blue',lwd=2,main="Weekend")
```



Weekday



Weekend