RR1

Huang,Zhen

2016年7月30日

Loading and preprocessing the data

We first load the data and briefly scan through the data:

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

```
tail(data)
```

```
##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

It can be found that there are some NAs in steps variable, the proportion of NAs in steps variable is:

```
sum(is.na(data$steps)) / length(data$steps)
```

```
## [1] 0.1311475
```

Actually in our preprocessing process we should deal with missing data, but since it is asked to be done later in this assignment, we will leave these NAs here.(It can be demonstrated that the other two variables do not have NA values)

```
sum(is.na(data$date))
```

```
## [1] 0
```

```
sum(is.na(data$interval))
```

```
## [1] 0
```

So now we do no transformations for the data.

What is mean total number of steps taken per day?

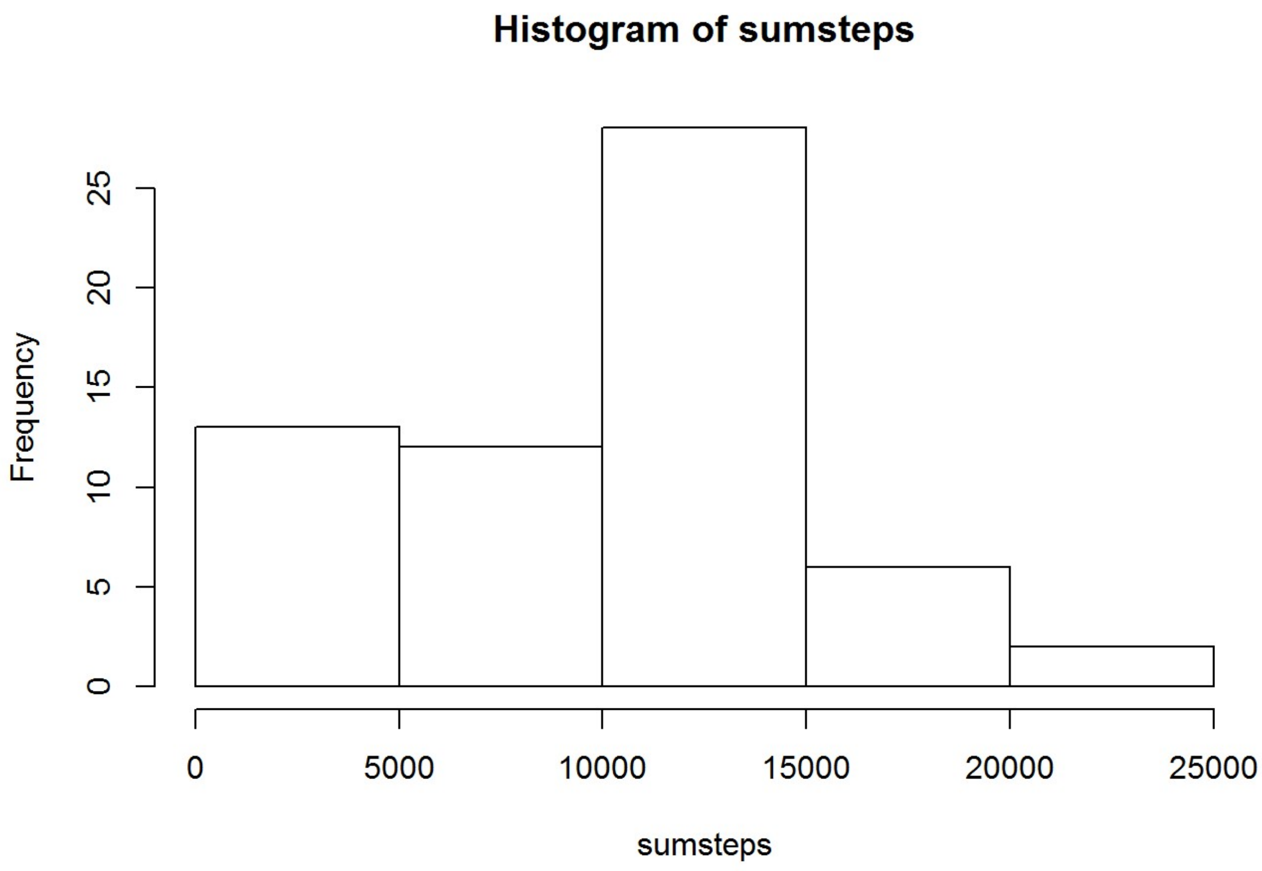
The total number of steps taken per day can be calculated as follows:

```
sumsteps <- tapply(data$steps, data$date, sum, na.rm = TRUE)
sumsteps
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06 
##          0        126      11352      12116      13294      15420 
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12 
##      11015          0      12811       9900      10304      17382 
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18 
##      12426      15098      10139      15084      13452      10056 
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24 
##      11829      10395       8821      13460       8918       8355 
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30 
##       2492       6778      10119      11458       5018       9819 
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05 
##      15414          0      10600      10571          0      10439 
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11 
##       8334      12883       3219          0          0      12608 
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17 
##      10765       7336          0         41       5441      14339 
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23 
##      15110       8841       4472      12787      20427      21194 
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29 
##      14478      11834      11162      13646      10183       7047 
## 2012-11-30 
##          0
```

A histogram summarizing the frequencies of sumsteps is shown below:

```
hist(sumsteps)
```



Also, the mean and median of sumsteps are presented here:

```
mean(sumsteps)
```

```
## [1] 9354.23
```

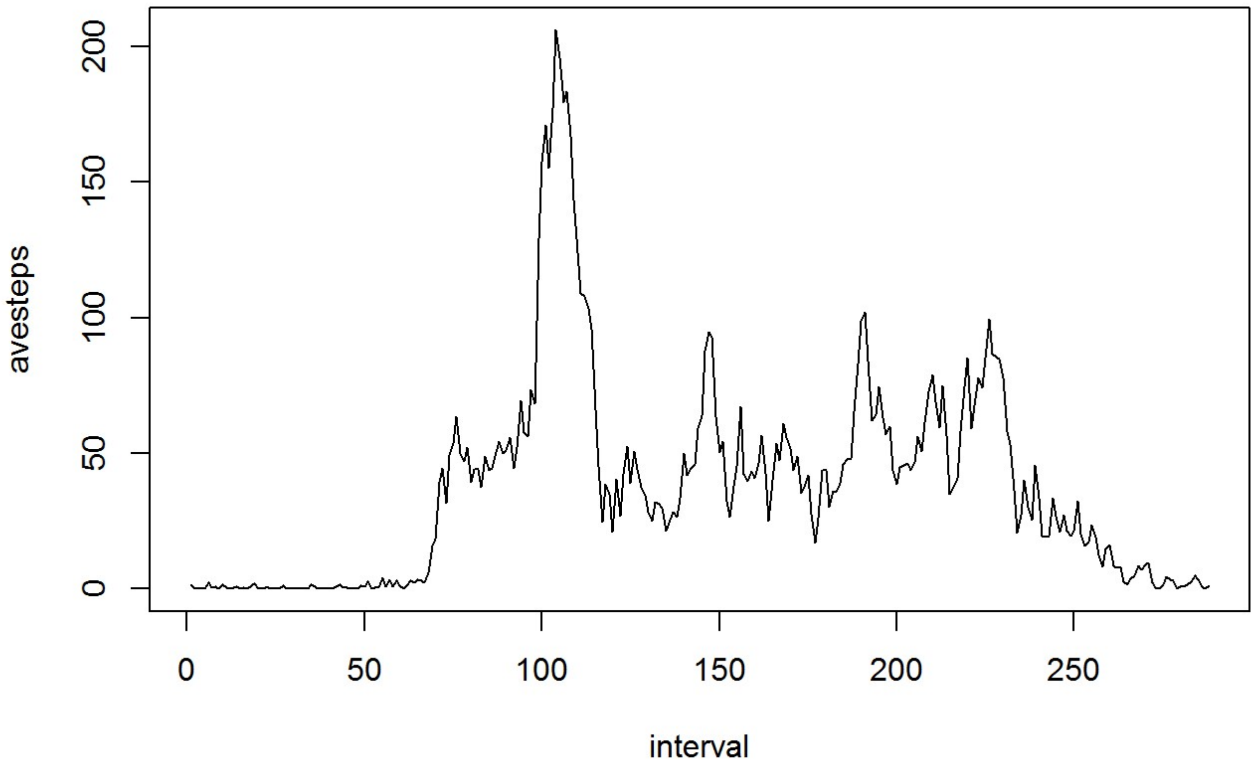
```
median(sumsteps)
```

```
## [1] 10395
```

What is the average daily activity pattern?

Average daily activity pattern:

```
avesteps <- tapply(data$steps, data$interval, mean, na.rm = TRUE)
plot(avesteps, type = "l", xlab = "interval")
```



Which contains the maximum number of steps:

```
max(avesteps, na.rm = TRUE)
```

```
## [1] 206.1698
```

```
names(which.max(avesteps)) ## which interval is the max
```

```
## [1] "835"
```

Imputing missing values

As demonstrated before, the total missing value number is:(We do not have missing values in date or interval)

```
sum(is.na(data$steps))
```

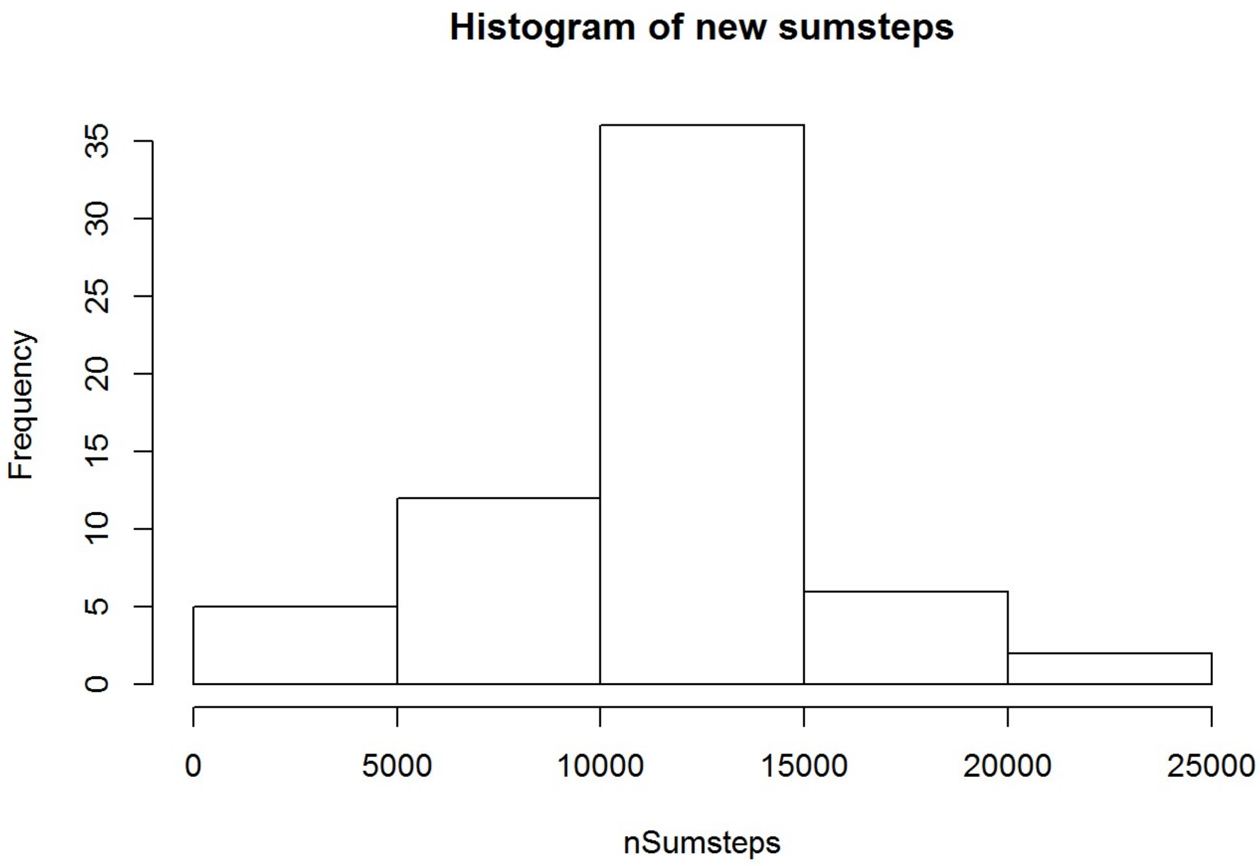
```
## [1] 2304
```

I use predictive mean matching(pmm method in mice) to impute missing data:

```
suppressMessages(library(mice))
nData <- complete(mice(data, m = 1, printFlag = FALSE))
```

Mean and median total number of steps are:

```
nSumsteps <- tapply(nData$steps, nData$date, sum, na.rm = TRUE)
hist(nSumsteps, main = "Histogram of new sumsteps")
```



```
mean(nSumsteps)
```

```
## [1] 11042.8
```

```
median(nSumsteps)
```

```
## [1] 11162
```

The mean and median are greater than the previous calculated values, and this is due to the contributions of new imputed values. The impacts of imputed values for each day is:

```
nSumsteps - sumsteps
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06 
##      12424          0          0          0          0          0 
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12 
##          0      14324          0          0          0          0 
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18 
##          0          0          0          0          0          0 
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24 
##          0          0          0          0          0          0 
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30 
##          0          0          0          0          0          0 
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05 
##          0      14441          0          0      14954          0 
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11 
##          0          0          0      10787      11857          0 
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17 
##          0          0      13142          0          0          0 
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23 
##          0          0          0          0          0          0 
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29 
##          0          0          0          0          0          0 
## 2012-11-30 
##      11074
```

The total impact is:

```
sum(nSumsteps - sumsteps)
```

```
## [1] 103003
```

Are there differences in activity patterns between weekdays and weekends?

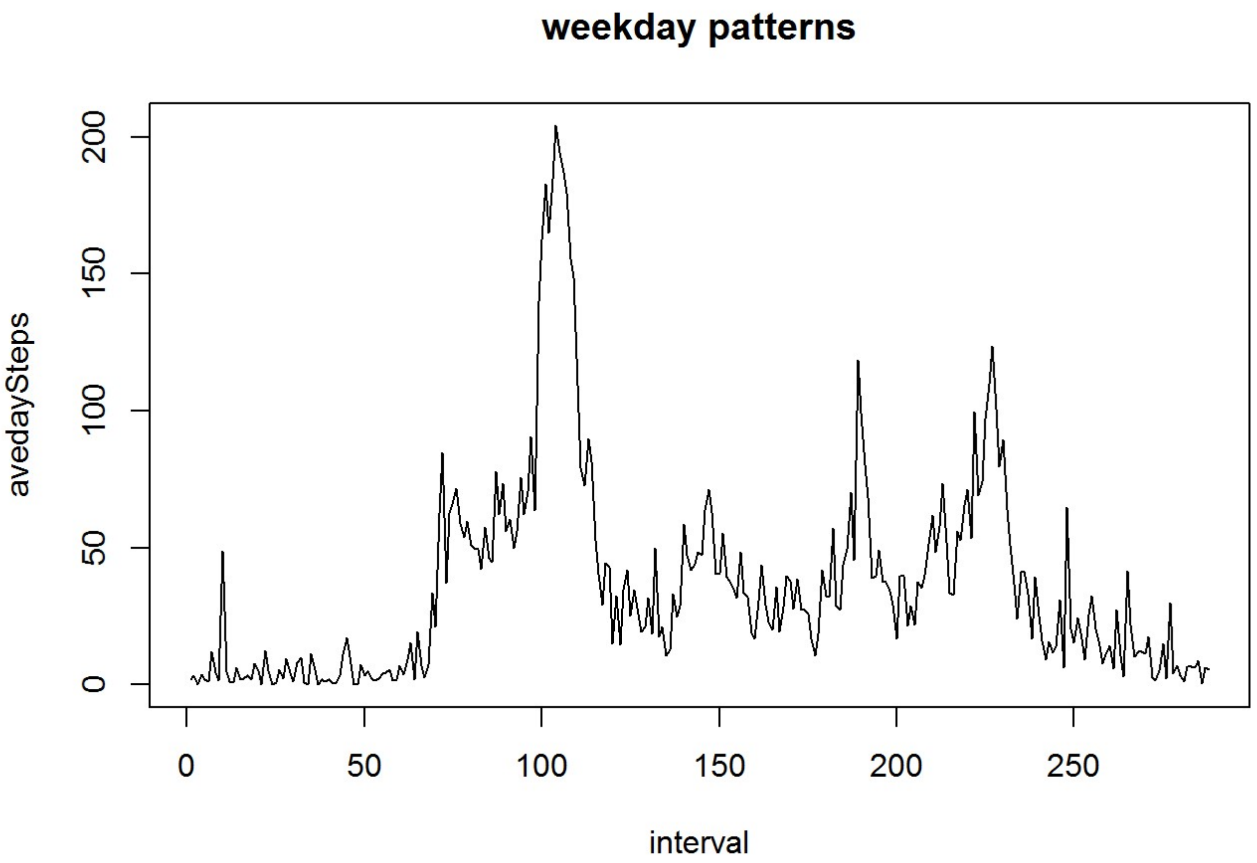
To answer this question, first we will have to generate a factor indicating whether a specific day is a weekday. The code is as below:

```
suppressMessages(library(chron))
isWeekday <- function(day) {
    if(is.weekend(day))
       return ("weekend")
    else
       return ("weekday")
}
nData$weekday <- sapply(nData$date, isWeekday)
head(nData) # show the structure of the new data
```

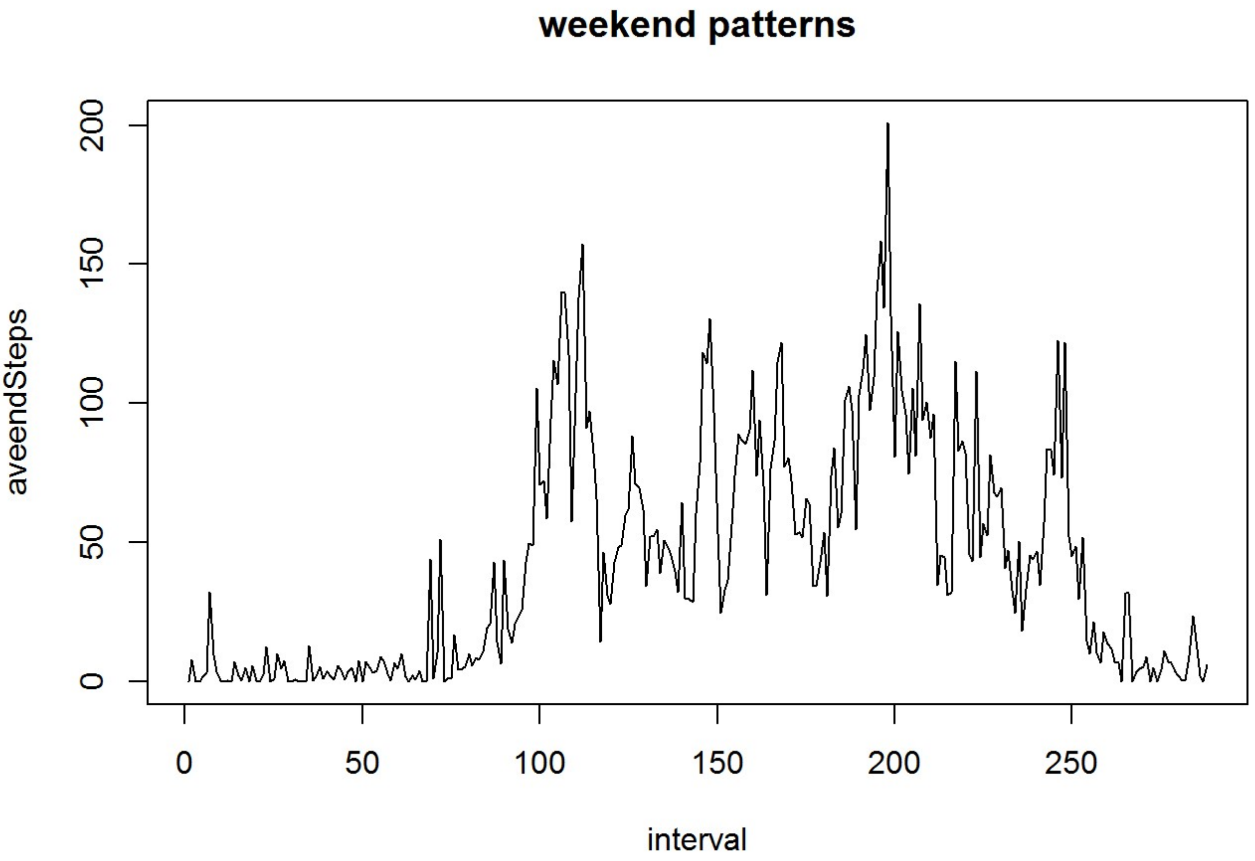
```
##   steps       date interval weekday
## 1     0 2012-10-01        0 weekday
## 2     0 2012-10-01        5 weekday
## 3     0 2012-10-01       10 weekday
## 4     0 2012-10-01       15 weekday
## 5    27 2012-10-01       20 weekday
## 6     0 2012-10-01       25 weekday
```

Then we can draw the plots:

```
weekdayData = nData[nData$weekday == "weekday",]
weekendData = nData[nData$weekday == "weekend",]
avedaySteps <- tapply(weekdayData$steps, weekdayData$interval, mean, na.rm = TRUE)
aveendSteps <- tapply(weekendData$steps, weekendData$interval, mean, na.rm = TRUE)
plot(avedaySteps, type = "l", xlab = "interval", main = "weekday patterns")
```



```
plot(aveendSteps, type = "l", xlab = "interval", main = "weekend patterns")
```


weekend patterns
aveendSteps
interval
0
50
100
150
200
0
50
100
150
200
250