Reproducible Research: Peer Assessment 1

Rafael Yukio Macena Yaguinuma

2022-09-05

activityData <- read.csv("./activity.csv")  
summary(activityData)

## steps date interval   
## Min. : 0.00 Length:17568 Min. : 0.0   
## 1st Qu.: 0.00 Class :character 1st Qu.: 588.8   
## Median : 0.00 Mode :character Median :1177.5   
## Mean : 37.38 Mean :1177.5   
## 3rd Qu.: 12.00 3rd Qu.:1766.2   
## Max. :806.00 Max. :2355.0   
## NA's :2304

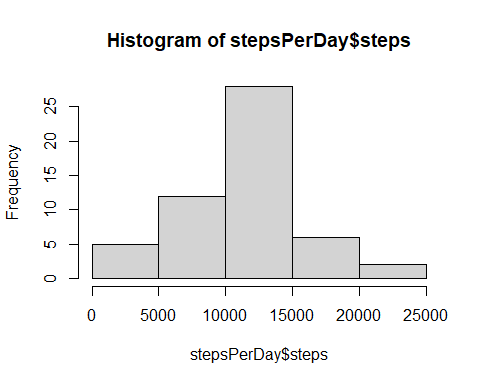
names(activityData)

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

head(activityData)

## 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

##Step 2 ##Histogram of the total number of steps taken each day  
  
stepsPerDay <- aggregate(steps ~ date, activityData, sum, na.rm=TRUE)  
hist(stepsPerDay$steps)



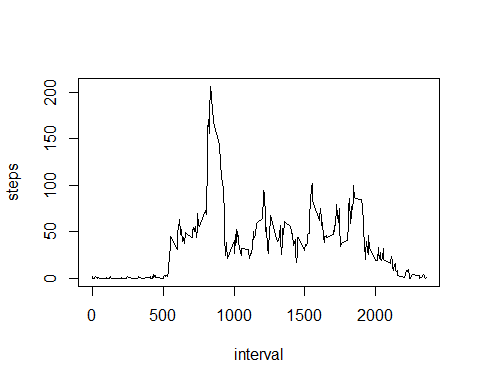
##Step 3 ##Mean and median number of steps taken each day  
  
meanStepsPerDay <- mean(stepsPerDay$steps)  
meanStepsPerDay

## [1] 10766.19

medianStepsPerDay <- median(stepsPerDay$steps)  
medianStepsPerDay

## [1] 10765

##Step 4 ##Time series plot of the average number of steps taken  
  
stepsPerInterval<-aggregate(steps~interval, data=activityData, mean, na.rm=TRUE)  
plot(steps~interval, data=stepsPerInterval, type="l")



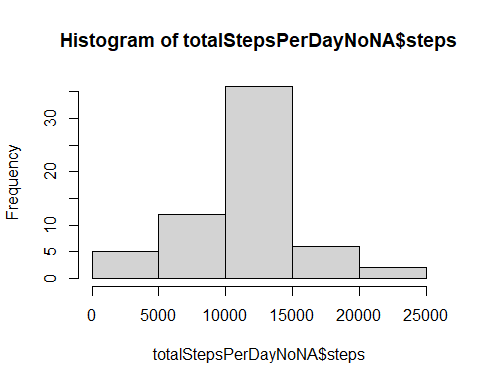
##Step 5 ##The 5-minute interval that, on average, contains the maximum number of steps  
  
intervalWithMaxNbSteps <- stepsPerInterval[which.max(stepsPerInterval$steps),]$interval  
intervalWithMaxNbSteps

## [1] 835

##Step 6 Code to describe and show a strategy for imputing missing data There are multiple strategies to deal with multiple value imputations.  
  
totalValuesMissings <- sum(is.na(activityData$steps))  
totalValuesMissings

## [1] 2304

getMeanStepsPerInterval<-function(interval){  
 stepsPerInterval[stepsPerInterval$interval==interval,]$steps  
}  
  
activityDataNoNA<-activityData  
for(i in 1:nrow(activityDataNoNA)){  
 if(is.na(activityDataNoNA[i,]$steps)){  
 activityDataNoNA[i,]$steps <- getMeanStepsPerInterval(activityDataNoNA[i,]$interval)  
 }  
}  
  
totalStepsPerDayNoNA <- aggregate(steps ~ date, data=activityDataNoNA, sum)  
hist(totalStepsPerDayNoNA$steps)  
  
meanStepsPerDayNoNA <- mean(totalStepsPerDayNoNA$steps)  
medianStepsPerDayNoNA <- median(totalStepsPerDayNoNA$steps)  
  
activityDataNoNA$date <- as.Date(strptime(activityDataNoNA$date, format="%Y-%m-%d"))  
activityDataNoNA$day <- weekdays(activityDataNoNA$date)  
for (i in 1:nrow(activityDataNoNA)) {  
 if (activityDataNoNA[i,]$day %in% c("Saturday","Sunday")) {  
 activityDataNoNA[i,]$day<-"weekend"  
 }  
 else{  
 activityDataNoNA[i,]$day<-"weekday"  
 }  
}  
stepsByDay <- aggregate(activityDataNoNA$steps ~ activityDataNoNA$interval + activityDataNoNA$day, activityDataNoNA, mean)  
  
names(stepsByDay) <- c("interval", "day", "steps")  
library(lattice)



xyplot(steps ~ interval | day, stepsByDay, type = "l", layout = c(1, 2),   
 xlab = "Interval", ylab = "Number of steps")

