Reproducible Research Project

Rizwan Ahmad

2 September 2020

## loading and reading the data

activity <- read.csv("activity.csv", stringsAsFactors=FALSE)  
str(activity)

## '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 ...

summary(activity)

## 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

Convert date to POSIXct class using lubridate package and convert interval to hour:minute format

library(lubridate)  
activity$date <- ymd(activity$date)  
str(activity)

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

### what is the average daily activity pattern?

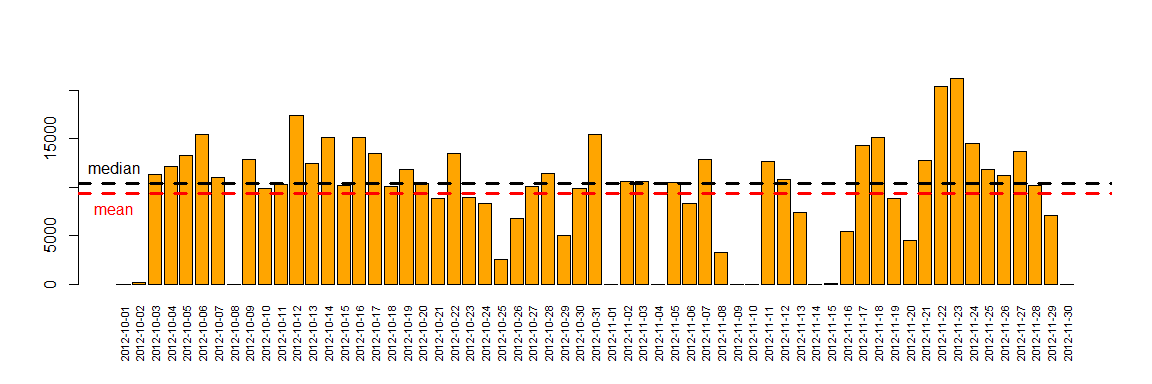
#### 1. Calculate the total number of steps taken per day (ignore the missing values)

require(dplyr)  
total\_day <- activity %>% group\_by(date)%>%summarise(total\_steps=sum(steps,na.rm=TRUE),na=mean(is.na(steps))) %>% print

## # A tibble: 61 x 3  
## date total\_steps na  
## <date> <int> <dbl>  
## 1 2012-10-01 0 1  
## 2 2012-10-02 126 0  
## 3 2012-10-03 11352 0  
## 4 2012-10-04 12116 0  
## 5 2012-10-05 13294 0  
## 6 2012-10-06 15420 0  
## 7 2012-10-07 11015 0  
## 8 2012-10-08 0 1  
## 9 2012-10-09 12811 0  
## 10 2012-10-10 9900 0  
## # ... with 51 more rows

`` visualize the total number of steps taken as barplot

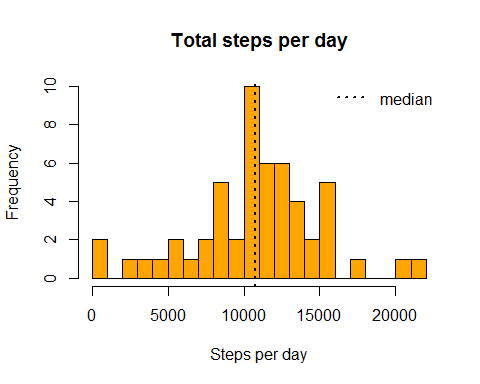
barplot(height = total\_day$total\_steps,names.arg=total\_day$date,cex.names=0.68,las=3,col="orange")  
abline(h=median(total\_day$total\_steps), lty=2,lwd=3, col="black")  
abline(h=mean(total\_day$total\_steps), lty=2,lwd=3, col="red")  
text(x = 0,y=median(total\_day$total\_steps),pos=3,labels = "median")  
text(x = 0,y=mean(total\_day$total\_steps),pos=1,labels = "mean",col="red")



### 2. Make a histogram of the total number of steps taken each day

Histogram does not contain days where all observations are missing (i.e. there have to be a number of steps for at least one interval for that day, to be included). Otherwise, there would be about ten days with 0 steps.

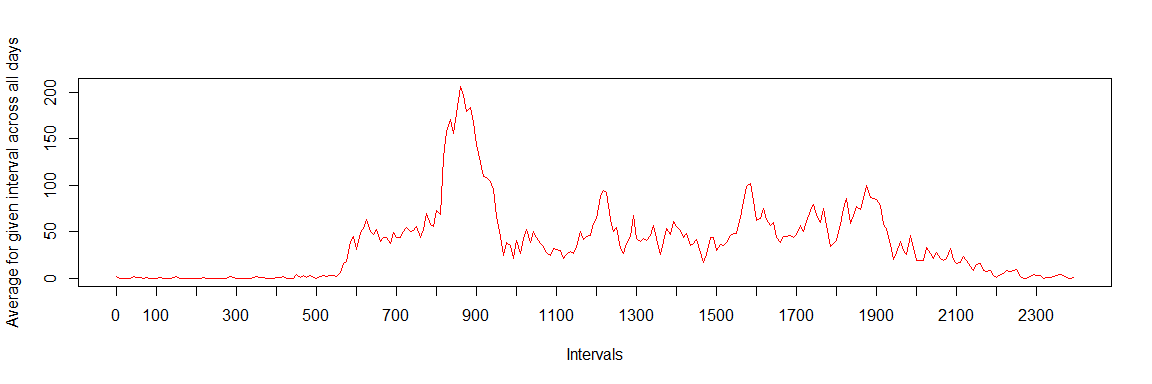
total\_day <- filter(total\_day, na < 1)  
hist(total\_day$total\_steps,col="orange",breaks=20,main="Total steps per day",xlab="Steps per day")  
abline(v=median(total\_day$total\_steps),lty=3, lwd=2, col="black")  
legend(legend="median","topright",lty=3,lwd=2,bty = "n")

 ### 3. Calculate and report the mean and median of the total number of steps taken per day

mean\_steps <- mean(total\_day$total\_steps,na.rm=TRUE)  
median\_steps <- median(total\_day$total\_steps,na.rm=TRUE)

Mean and median of the total number of steps taken per day are 1.07661910^{4} steps and 10765 steps, respectively. ## What is the average daily activity pattern? ### 1. Make 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 days (y-axis)

library(dplyr,quietly = TRUE)  
daily\_patterns <- activity %>% group\_by(interval) %>% summarise(average=mean(steps,na.rm=TRUE))  
plot(x = 1:nrow(daily\_patterns),y = daily\_patterns$average,type = "l",  
 col = "red", xaxt = "n",xlab="Intervals",   
 ylab = "Average for given interval across all days")  
axis(1,labels=daily\_patterns$interval[seq(1,288,12)],  
 at = seq\_along(daily\_patterns$interval)[seq(1,288,12)])



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

max\_numb\_steps\_interval <- filter(daily\_patterns,average==max(average))

Interval **“835”** contains on average the maximum number of steps (**206.17 steps**).

## Imputing missing values

#### 1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

na\_number <- sum(is.na(activity$steps))  
na\_number

## [1] 2304

percentage\_na <- mean(is.na(activity$steps))  
percentage\_na

## [1] 0.1311475

Total number of missing values in the dataset amounts to **2304**  (what is **13.1** % of total observations).

#### 2. Devise a strategy for filling in all of the missing values in the dataset

As the number of missing values in this dataset is fairly large, we cannot be sure if there is no bias introduced by missing values. Therefore we impute missing values based on average number of steps in particular 5-minutes interval.

#### 3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

without\_NAs <- numeric(nrow(activity))  
for (i in 1:nrow(activity))  
{  
 if (is.na(activity[i,"steps"])==TRUE)  
 {  
 without\_NAs[i]<-filter(daily\_patterns,interval==activity[i,"interval"]) %>% select(average)  
 }   
 else  
 {  
 without\_NAs[i]<-activity[i,"steps"]  
 }  
   
}  
activity\_without\_NAs<-mutate(activity,steps\_no\_NAs=without\_NAs)  
head(activity\_without\_NAs)

## steps date interval steps\_no\_NAs  
## 1 NA 2012-10-01 0 1.716981  
## 2 NA 2012-10-01 5 0.3396226  
## 3 NA 2012-10-01 10 0.1320755  
## 4 NA 2012-10-01 15 0.1509434  
## 5 NA 2012-10-01 20 0.0754717  
## 6 NA 2012-10-01 25 2.09434

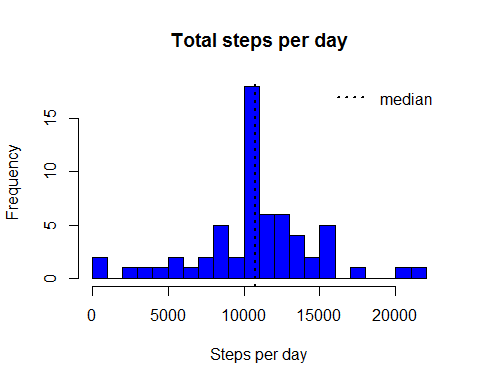
Below code is just to verify if process of imputing missing values correctly preserved original values (lines with no NAs)

check <- filter(activity\_without\_NAs,!is.na(steps)) %>% mutate(ok = (steps==steps\_no\_NAs))  
mean(check$ok)

## [1] 1

#### 4. Make 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

total\_day\_noNAs <- activity\_without\_NAs %>% mutate(steps\_no\_NAs=as.numeric(steps\_no\_NAs)) %>% group\_by(date) %>% summarise(total\_steps=sum(steps\_no\_NAs))  
hist(total\_day\_noNAs$total\_steps,col="blue",breaks=20,main="Total steps per day",xlab="Steps per day")  
abline(v=median(total\_day$total\_steps),lty=3, lwd=2, col="black")  
legend(legend="median","topright",lty=3,lwd=2,bty = "n")



summary(total\_day\_noNAs$total\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41 9819 10766 10766 12811 21194

Imputing missing values, mean of the total number of steps taken per day increased while median decreased,compared to estimates from the first part (ingoring missing values). Imputing missing data resulted in increase of total daily number of steps (instead of each NAs we have average that is always >=0)

## Are there differences in activity patterns between weekdays and weekends?

#### 1. Create a new factor variable in the dataset with two levels - “weekday” and “weekend” indicating whether a given date is a weekday or weekend day

library(lubridate)  
is\_weekday <-function(date){  
 if(wday(date)%in%c(1,7)) result<-"weekend"  
 else  
 result<-"weekday"  
 result  
}  
activity\_without\_NAs <- mutate(activity\_without\_NAs,date=ymd(date)) %>% mutate(day=sapply(date,is\_weekday))  
table(activity\_without\_NAs$day)

##   
## weekday weekend   
## 12960 4608

#### 2. Make 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)

library(ggplot2)  
daily\_patterns <- activity\_without\_NAs %>% mutate(day=factor(day,levels=c("weekend","weekday")),steps\_no\_NAs=as.numeric(steps\_no\_NAs)) %>% group\_by(interval,day) %>% summarise(average=mean(steps\_no\_NAs))  
qplot(interval,average,data=daily\_patterns,geom="line",facets=day~.)

