Reproducible Research: Peer Assessment 1

```
### Load the lattice package for plotting some graphs
library(lattice)
```

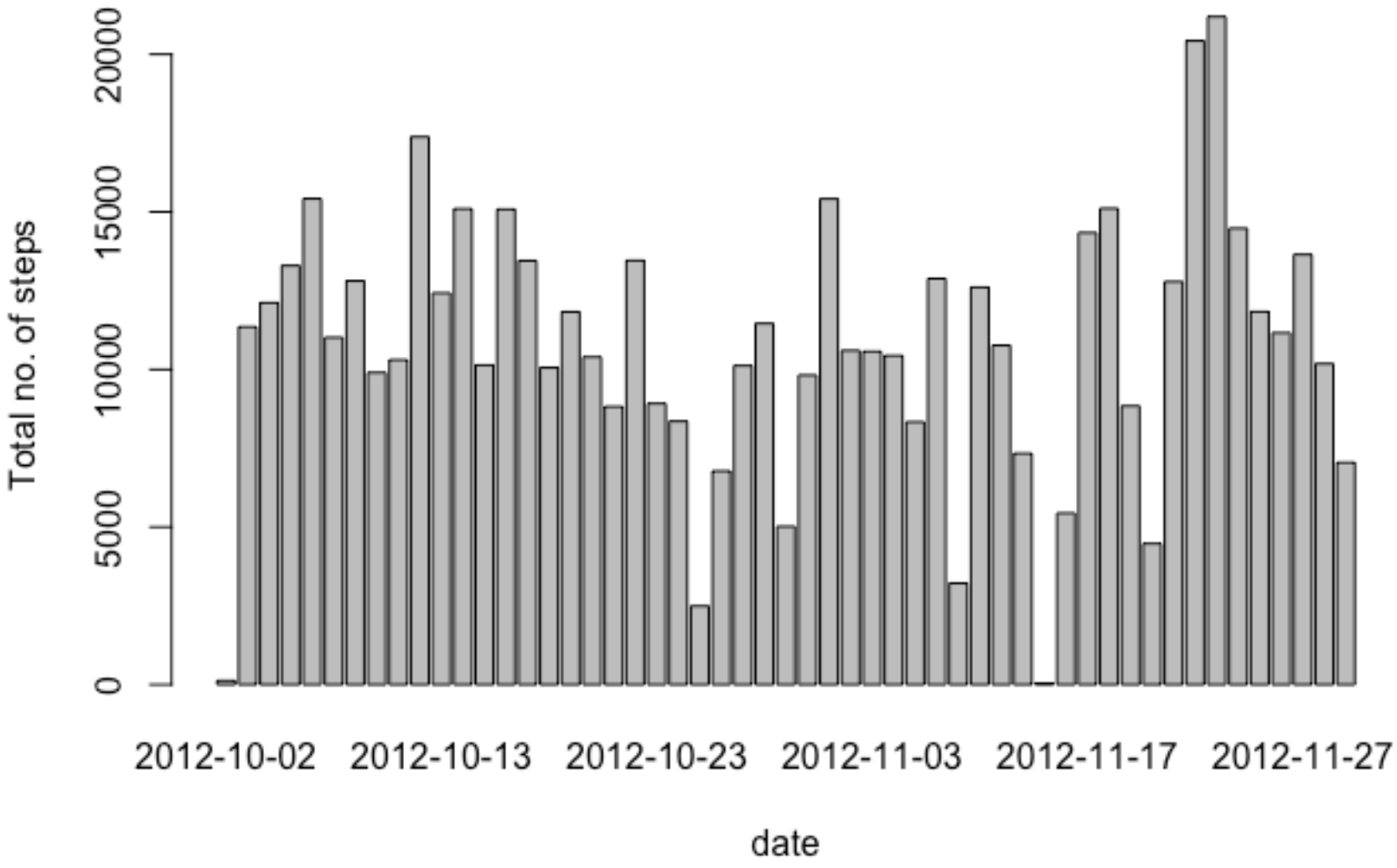
Loading and preprocessing the data

```
### Read the original data
origData <- read.csv("activity.csv", header=TRUE, na.strings = "NA")

### Remove NA cases
completeData <- origData[complete.cases(origData),]
### Convert the dates from strings
completeData$date <- as.Date(completeData$date)
head(completeData)
```

```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
```

What is mean total number of steps taken per day?

```
### Find out total no. of steps taken per day
totals <- tapply(completeData$steps, completeData$date, FUN=sum, na.rm=TRUE)

### What is the average daily activity pattern?
barplot(totals, xlab="date", ylab="Total no. of steps")
```




```
### Mean and median
mean(totals)
```

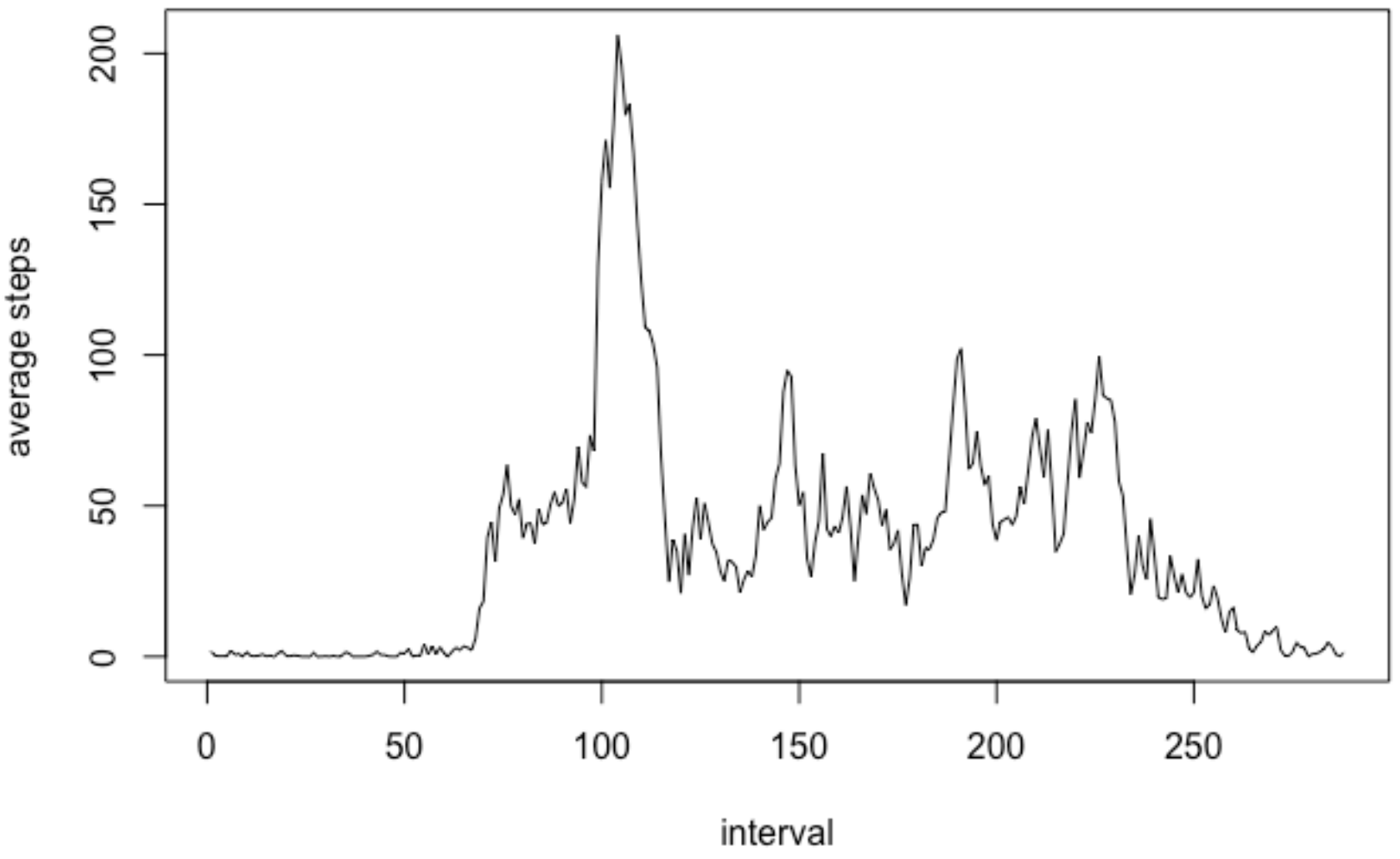
```
## [1] 10766.19
```

```
median(totals)
```

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

What is the average daily activity pattern?

```
averages <- tapply(completeData$steps, completeData$interval, FUN=mean, na.rm=TRUE
)
plot(averages, type="l", xlab="interval", ylab="average steps")
```



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

```
names(averages[averages==max(averages)])
```

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

Imputing missing values

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

```
nrow(origData) - nrow(completeData)
```

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

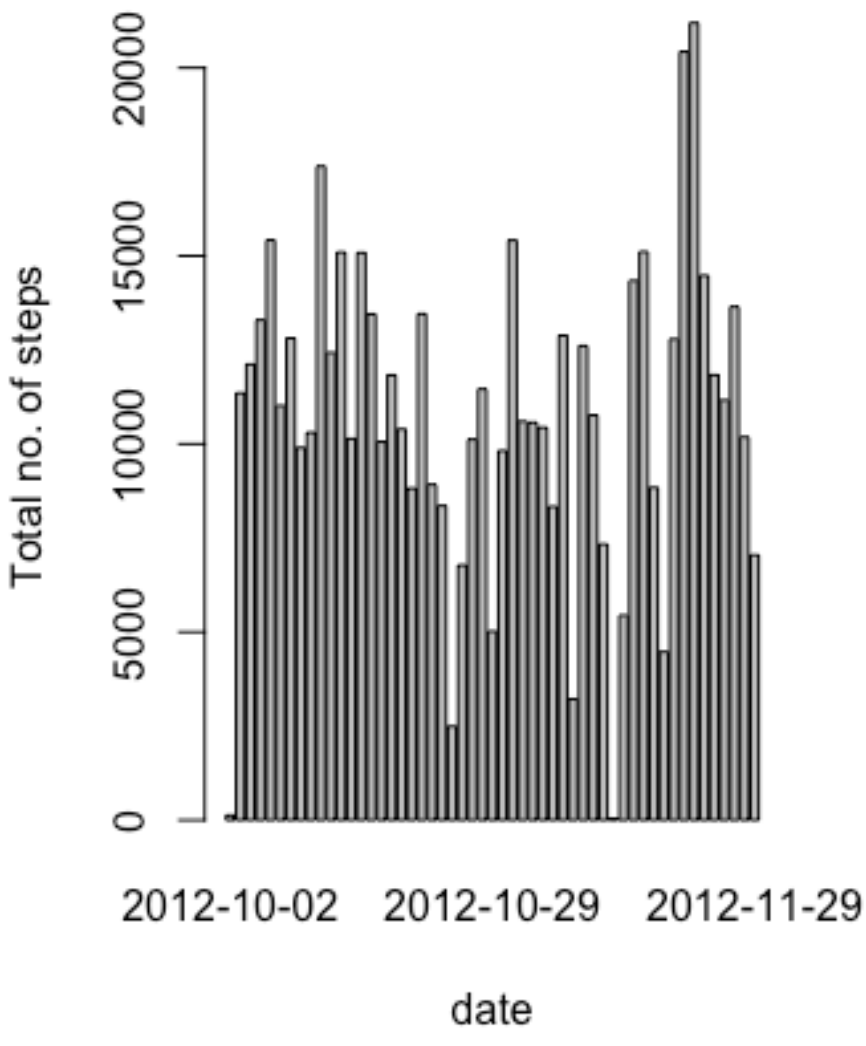
Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

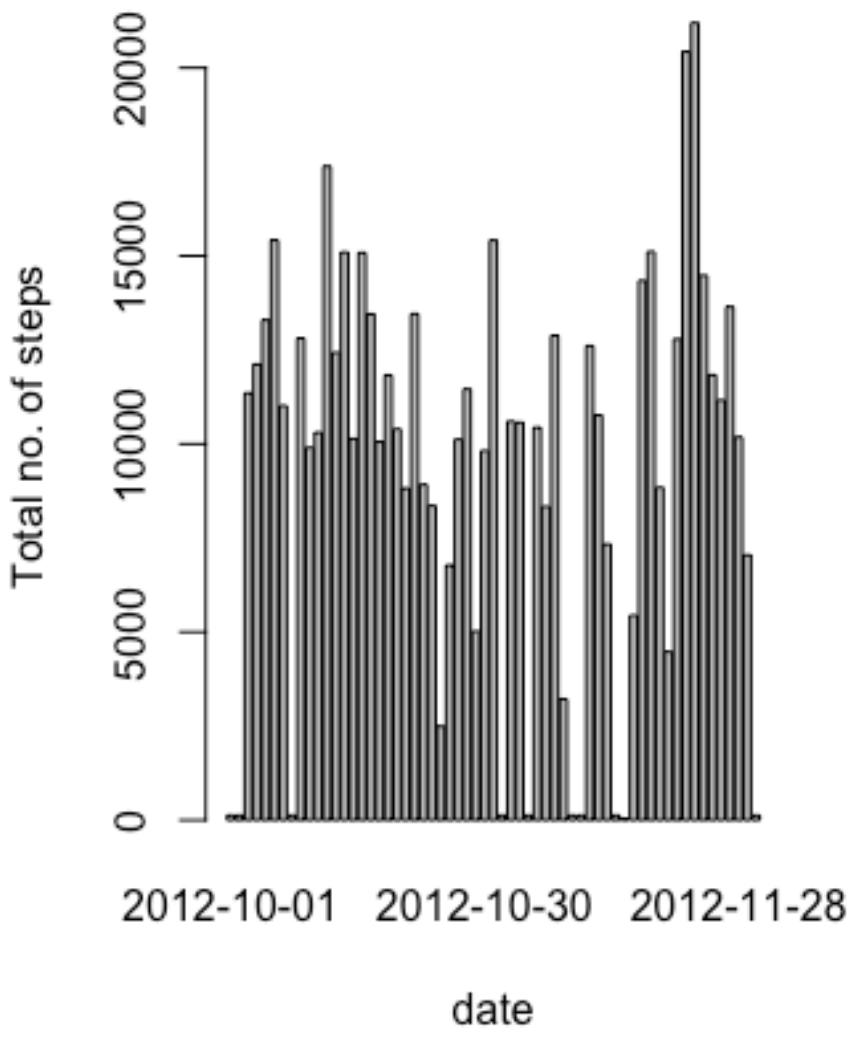
Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
### Create a new dataset
newData <- origData

### For each day, find the average no. of steps
### Then fill into the rows with NA
for(date in names(totals)) {
        dailyAverage <- mean(origData[origData$date == date,]$steps)
        newData[!complete.cases(newData), "steps"] <- dailyAverage
}
```

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

```
### Find out total no. of steps taken per day from the new data set
par(mfrow=c(1,2))

### Plot old graph
totals <- tapply(completeData$steps, completeData$date, FUN=sum, na.rm=TRUE)
barplot(totals, main="Before imputing missing values", xlab="date", ylab="Total no
. of steps")

### Plot new graph
newtotals <- tapply(newData$steps, newData$date, FUN=sum, na.rm=TRUE)
barplot(newtotals, main="After imputing missing values", xlab="date", ylab="Total
no. of steps")
```





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

```
mean(newtotals)
```

```
## [1] 9370.754
```

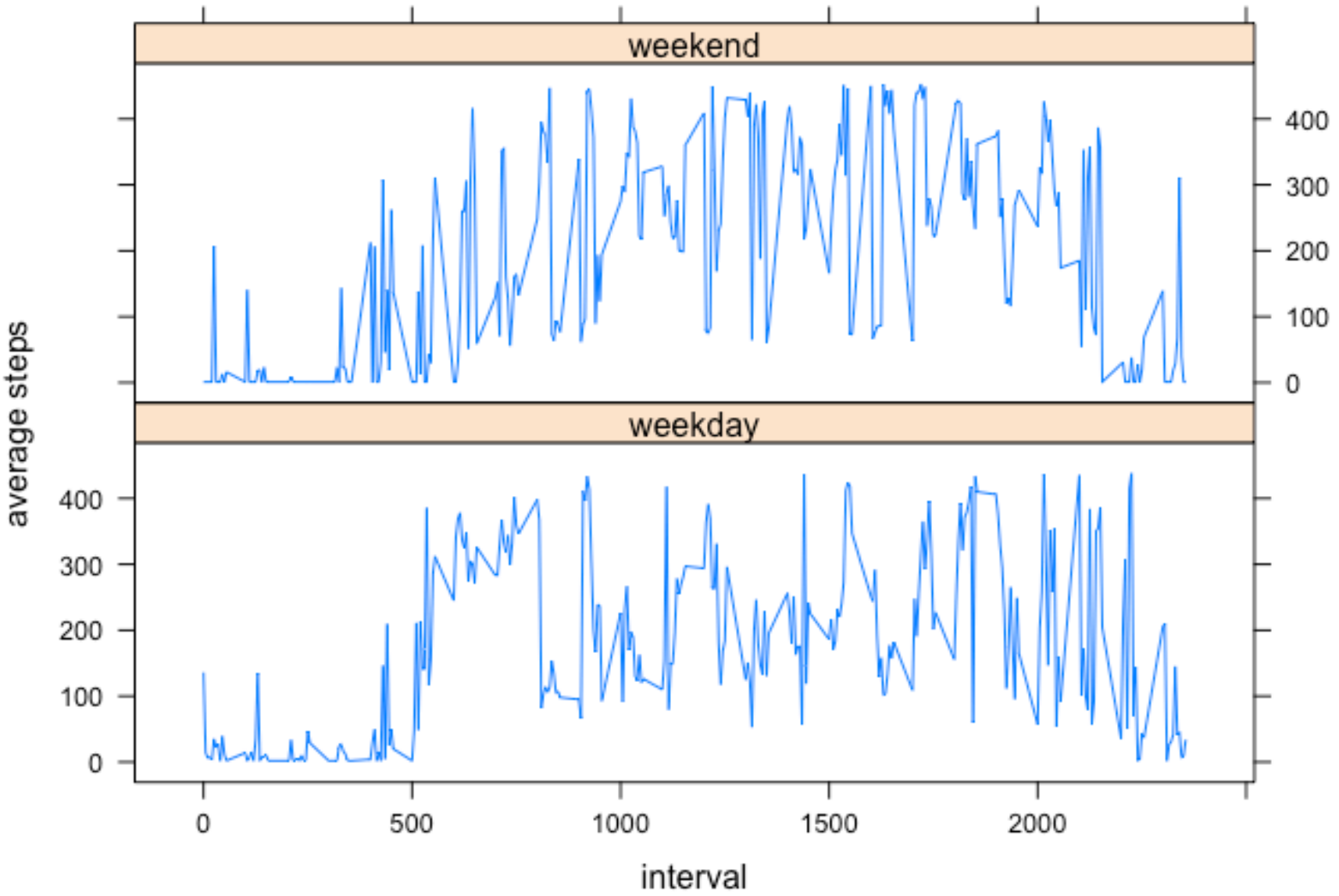
```
median(newtotals)
```

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

```
## Ans: Yes, the mean & median are decreased.
```

Are there differences in activity patterns between weekdays and weekends?

```
### Assign the "dayType" to "weekend" if that day is Saturday or Sunday
newData$day <- weekdays(as.Date(newData$date))
newData[newData$day == "Saturday", "dayType"] <- "weekend"
newData[newData$day == "Sunday", "dayType"] <- "weekend"
### Otherwise it's a "weekday""
newData[is.na(newData$dayType), "dayType"] <- "weekday"

### Separate between weekday & weekend data
weekdayData <- newData[newData$dayType == "weekday",]
weekendData <- newData[newData$dayType == "weekend",]

### For each interval, find out the averages
weekdayAverage <- tapply(weekdayData$steps, weekdayData$interval, FUN=mean)
weekendAverage <- tapply(weekendData$steps, weekendData$interval, FUN=mean)

### Construct a big data frame first that contains the average, interval & day-typ
e
overallAverage <- as.data.frame(rbind(cbind(names(weekdayAverage), weekdayAverage,
"weekday"),
                        cbind(names(weekendAverage), weekendAverage, "weekend")))
### Assign the column names
colnames(overallAverage) <- c("interval", "average", "daytype")

### Output the panel plots
xyplot(as.numeric(average) ~ as.numeric(as.character(interval)) | daytype, data=ov
erallAverage, type='l', layout=c(1, 2), xlab="interval", ylab="average steps")
```


weekend
weekday
average steps
0
100
200
300
400
0
500
1000
1500
2000
interval