PA1_template

Mohamed Ali

May 21, 2016

Reproducible Research: WEEK2: Peer Assessment 1

Loading and preprocessing the data

Downloading data and reading it using read.csv():

```
filename <- "repdata-data-activity.zip"
if (!file.exists(filename)){
      download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Fact
ivity.zip",
                    destfile = filename, method = "wget")
      unzip(filename)
}
activityData <- read.csv("activity.csv")
```

2. Process/transform the data (if necessary) into a format suitable for your analysis

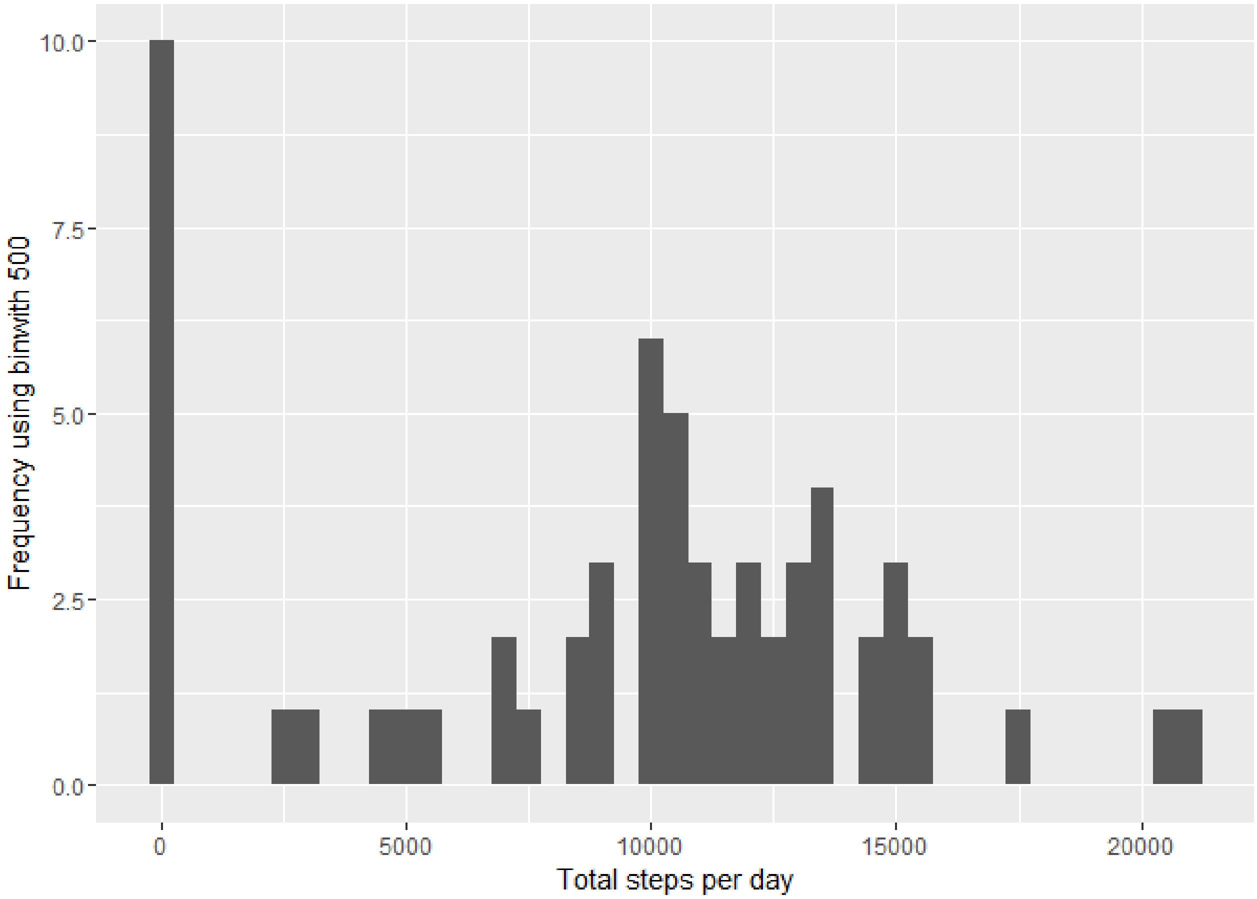
```
#activityData$interval <- strptime(gsub("([0-9]{1,2})([0-9]{2})", "\\1:\\2", ac
tivityData$interval), format='%H:%M')
```

What is mean total number of steps taken per day?

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

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

```
qplot(stepsByDay, xlab='Total steps per day', ylab='Frequency using binwith 50
0', binwidth=500)
```



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

```
stepsByDayMean <- mean(stepsByDay)
stepsByDayMedian <- median(stepsByDay)
```

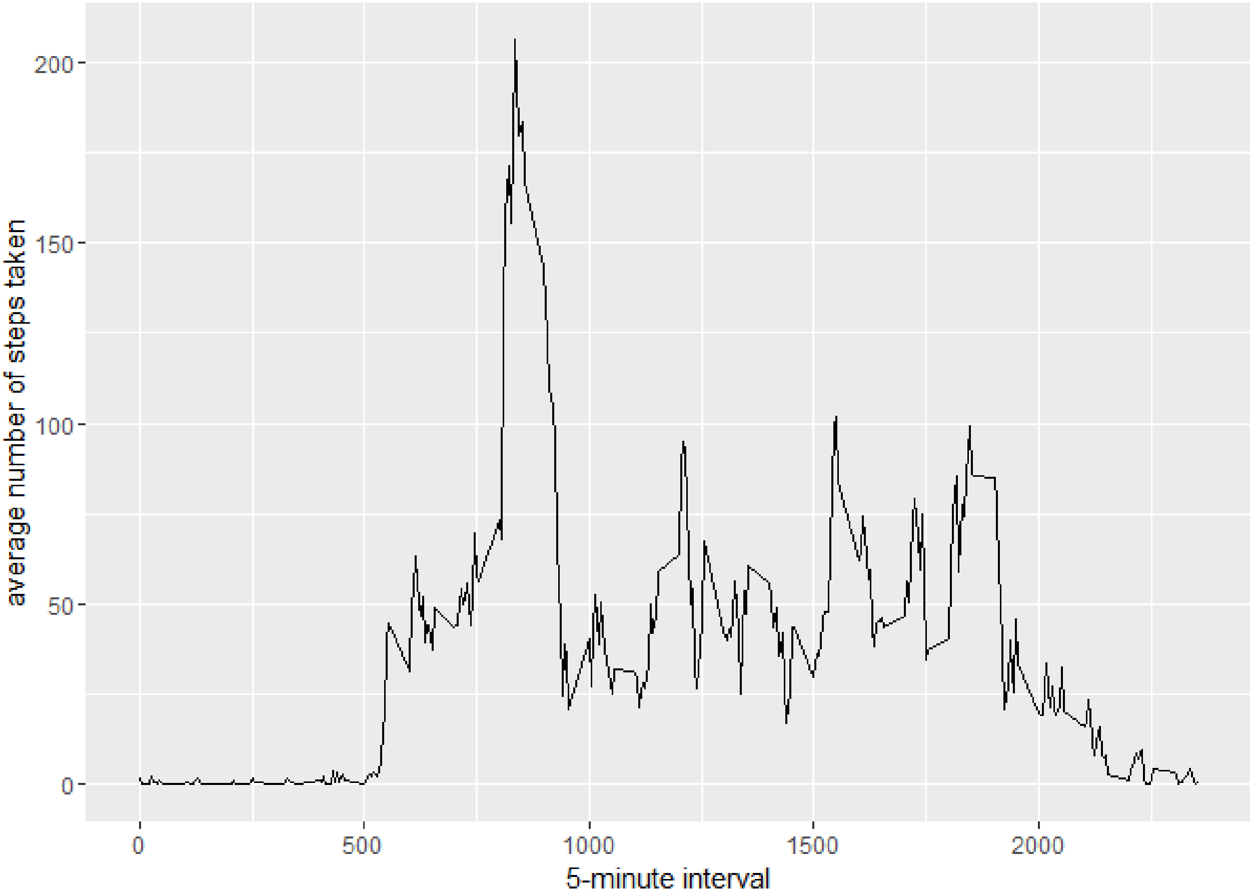
- Mean: 9354.2295082
- Median: 10395

What is the average daily activity pattern?

```
averageStepsPerTimeBlock <- aggregate(x=list(meanSteps=activityData$steps), by=
list(interval=activityData$interval), FUN=mean, na.rm=TRUE)
```

1. Make a time series plot

```
ggplot(data=averageStepsPerTimeBlock, aes(x=interval, y=meanSteps)) +
    geom_line() +
    xlab("5-minute interval") +
    ylab("average number of steps taken")
```



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

```
mostSteps <- which.max(averageStepsPerTimeBlock$meanSteps)
timeMostSteps <-  gsub("([0-9]{1,2})([0-9]{2})", "\\1:\\2", averageStepsPerTimeBlock[mostSteps,'interval'])
```

- Most Steps at: 8:35

Imputing missing values

1. Calculate and report the total number of missing values in the dataset

```
numMissingValues <- length(which(is.na(activityData$steps)))
```

- Number of missing values: 2304

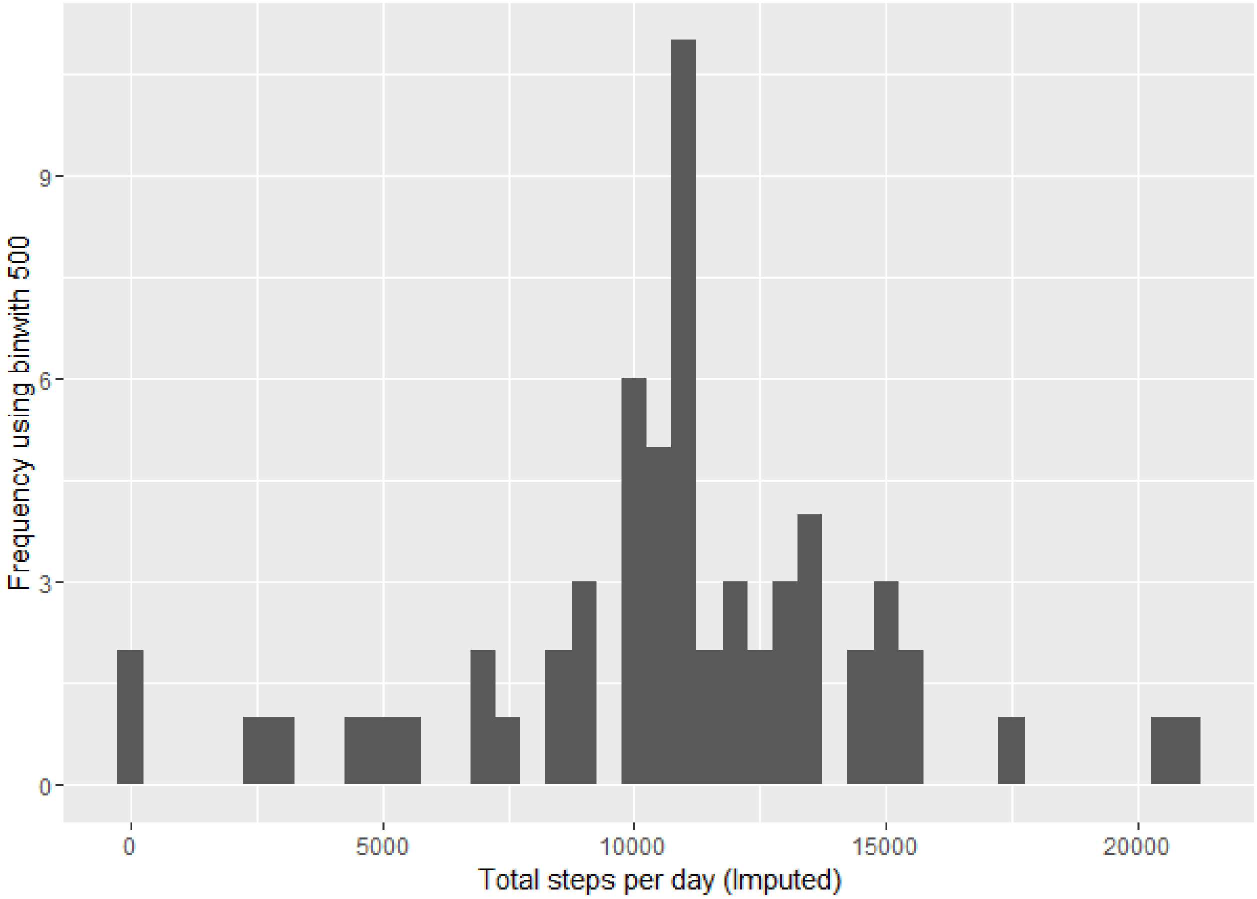
2. Devise a strategy for filling in all of the missing values in the dataset.

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

```
activityDataImputed <- activityData
activityDataImputed$steps <- impute(activityData$steps, fun=mean)
```

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

```
stepsByDayImputed <- tapply(activityDataImputed$steps, activityDataImputed$dat
e, sum)
qplot(stepsByDayImputed, xlab='Total steps per day (Imputed)', ylab='Frequency
using binwith 500', binwidth=500)
```



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

```
stepsByDayMeanImputed <- mean(stepsByDayImputed)
stepsByDayMedianImputed <- median(stepsByDayImputed)
```

- Mean (Imputed): 1.076618910^{4}
- Median (Imputed): 1.076618910^{4}

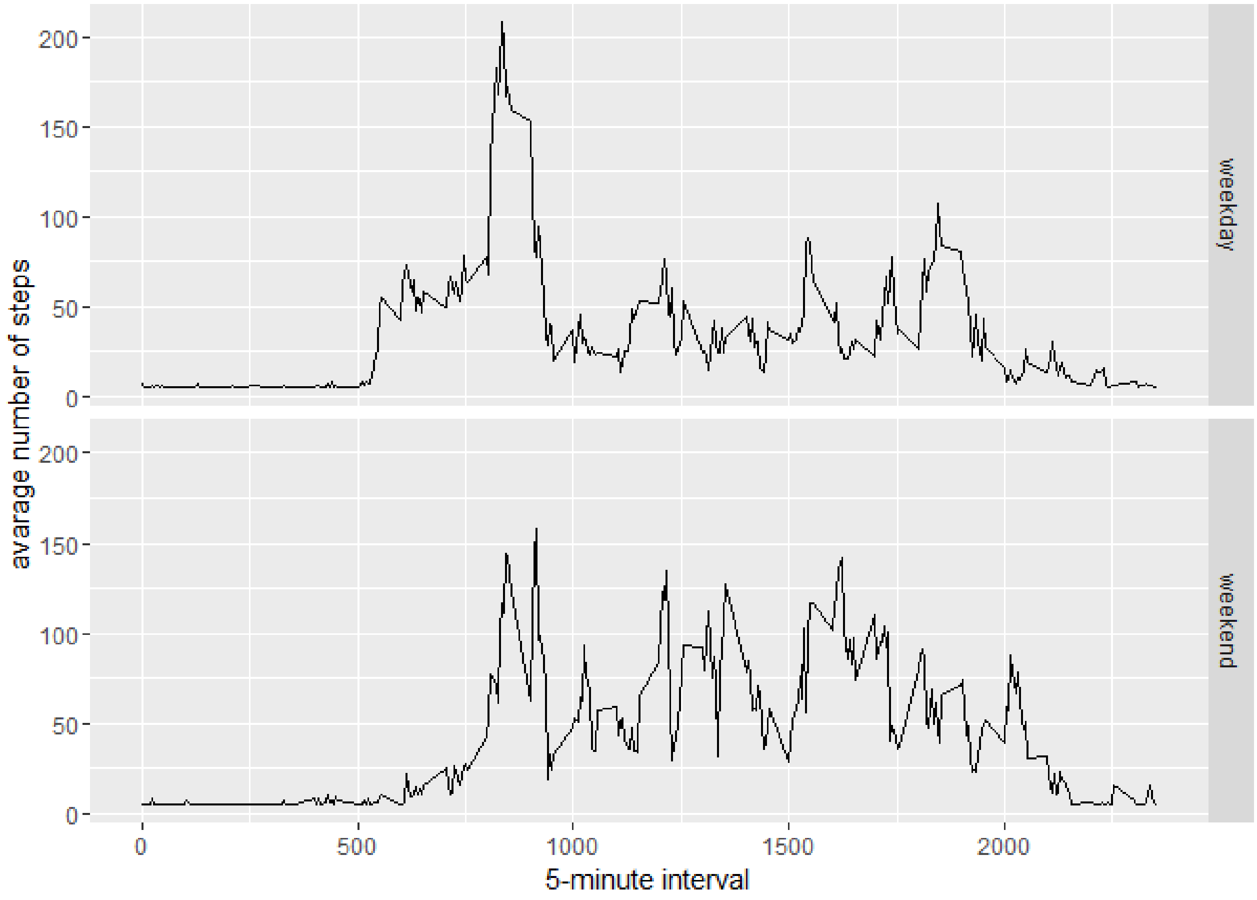
Are there differences in activity patterns between weekdays and weekends?

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.

```
activityDataImputed$dateType <-  ifelse(as.POSIXlt(activityDataImputed$date)$wd
ay %in% c(0,6), 'weekend', 'weekday')
```

2. Make a panel plot containing a time series plot

```
averagedActivityDataImputed <- aggregate(steps ~ interval + dateType, data=acti
vityDataImputed, mean)
ggplot(averagedActivityDataImputed, aes(interval, steps)) +
    geom_line() +
    facet_grid(dateType ~ .) +
    xlab("5-minute interval") +
    ylab("avarage number of steps")
```


avarage number of steps
weekday
weekend
5-minute interval