Reproducible Research Peer Assessment 1

gpalani

June 14, 2015

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the “quantified self” movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

Data

The data for this assignment can be downloaded from the course web site:

Dataset: Activity monitoring data [52K] The variables included in this dataset are:

steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)

date: The date on which the measurement was taken in YYYY-MM-DD format

interval: Identifier for the 5-minute interval in which measurement was taken

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

Loading and preprocessing the data

Show any code that is needed to

1. Load the data (i.e. read.csv())
2. Process/transform the data (if necessary) into a format suitable for your analysis

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

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day

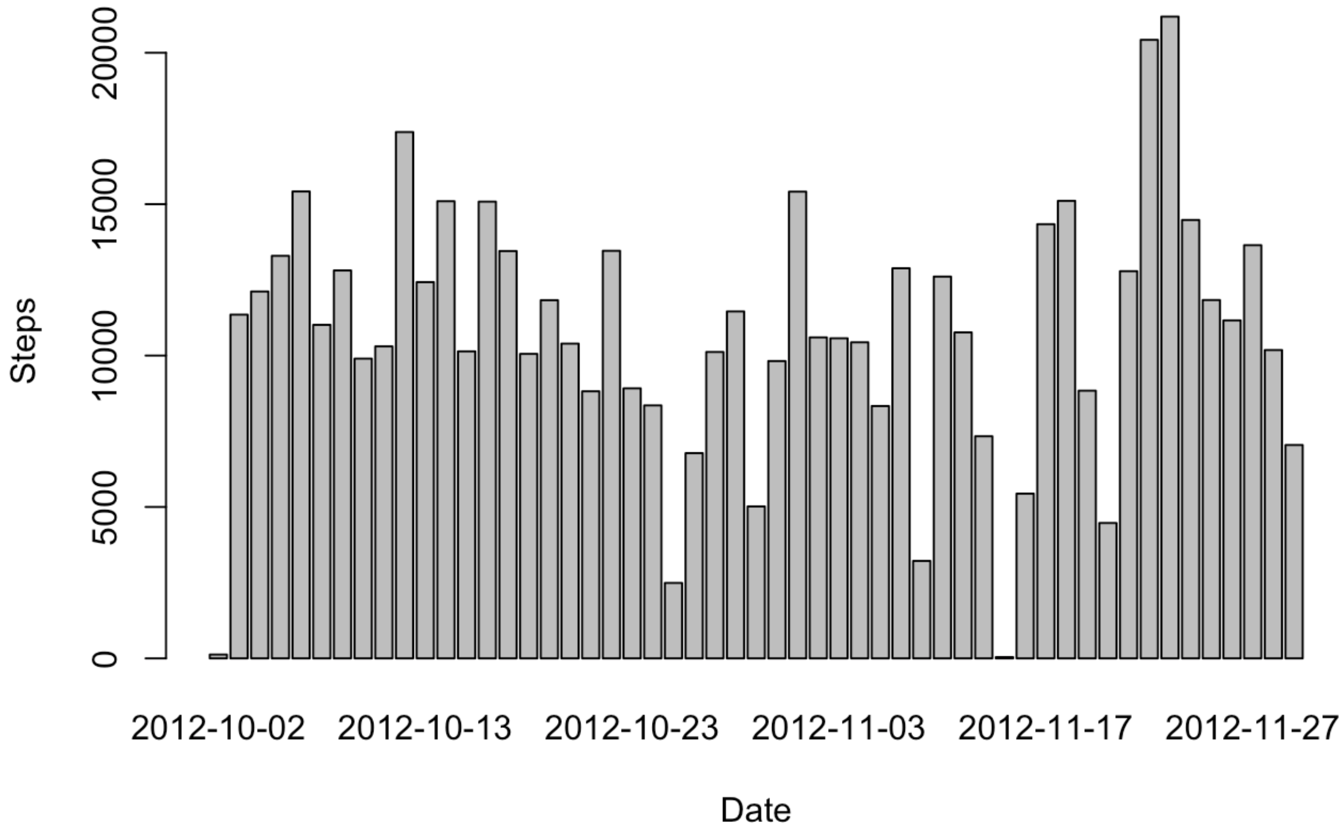
    ```
    Calculating and outputing the first 20 rows
    ```

```
stepsData<- aggregate(steps ~ date, data = activityData, FUN = sum, na.rm = T)
head(stepsData,20)
```

```
##          date steps
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## 11 2012-10-13 12426
## 12 2012-10-14 15098
## 13 2012-10-15 10139
## 14 2012-10-16 15084
## 15 2012-10-17 13452
## 16 2012-10-18 10056
## 17 2012-10-19 11829
## 18 2012-10-20 10395
## 19 2012-10-21  8821
## 20 2012-10-22 13460
```

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

```
barplot(stepsData$steps, names.arg = stepsData$date, xlab = "Date", ylab = "Steps")
```



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

Mean

```
print(sprintf("Mean total steps taken per day: %f ",mean(stepsData$steps, na.rm = T)))
```

```
## [1] "Mean total steps taken per day: 10766.188679 "
```

Median

```
print(sprintf("Median total steps taken per day: %f ",median(stepsData$steps, na.rm = T)))
```

```
## [1] "Median total steps taken per day: 10765.000000 "
```

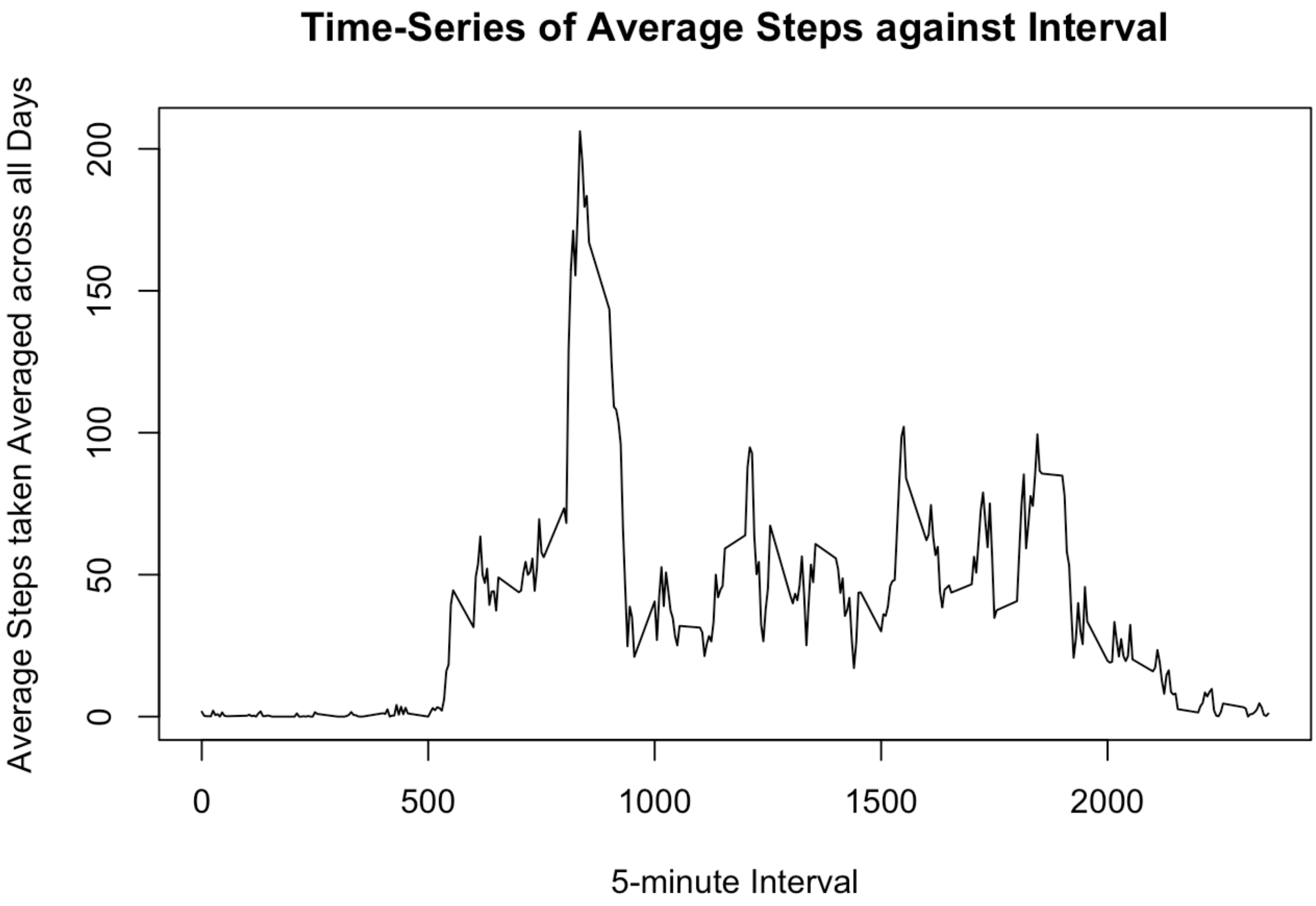
What is the average daily activity pattern?

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)

```
minIntervalSteps <- aggregate(
    data=activityData,
    steps~interval,
    FUN=mean,
    na.action=na.omit
)
colnames(minIntervalSteps) <- c("minInterval", "avgStepsAcrossDay")
```

Plotting the time series plot…

```
with(minIntervalSteps, {
    plot(
      x=minInterval,
      y=avgStepsAcrossDay,
      type="l",
      main="Time-Series of Average Steps against Interval",
      xlab="5-minute Interval",
      ylab="Average Steps taken Averaged across all Days"

    )
})
```



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

   ```
   Finding the 5-minute interval with the maximum number of steps
   ```

```
intervalMax <- minIntervalSteps[minIntervalSteps$avgStepsAcrossDay==max(minIntervalSteps$avgS
tepsAcrossDay),]
intervalMax$minInterval
```

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

Inputting missing values

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

```
print(sprintf("Total number of rows : %f ",sum(is.na(activityData))))
```

```
## [1] "Total number of rows : 2304.000000 "
```

2. 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.

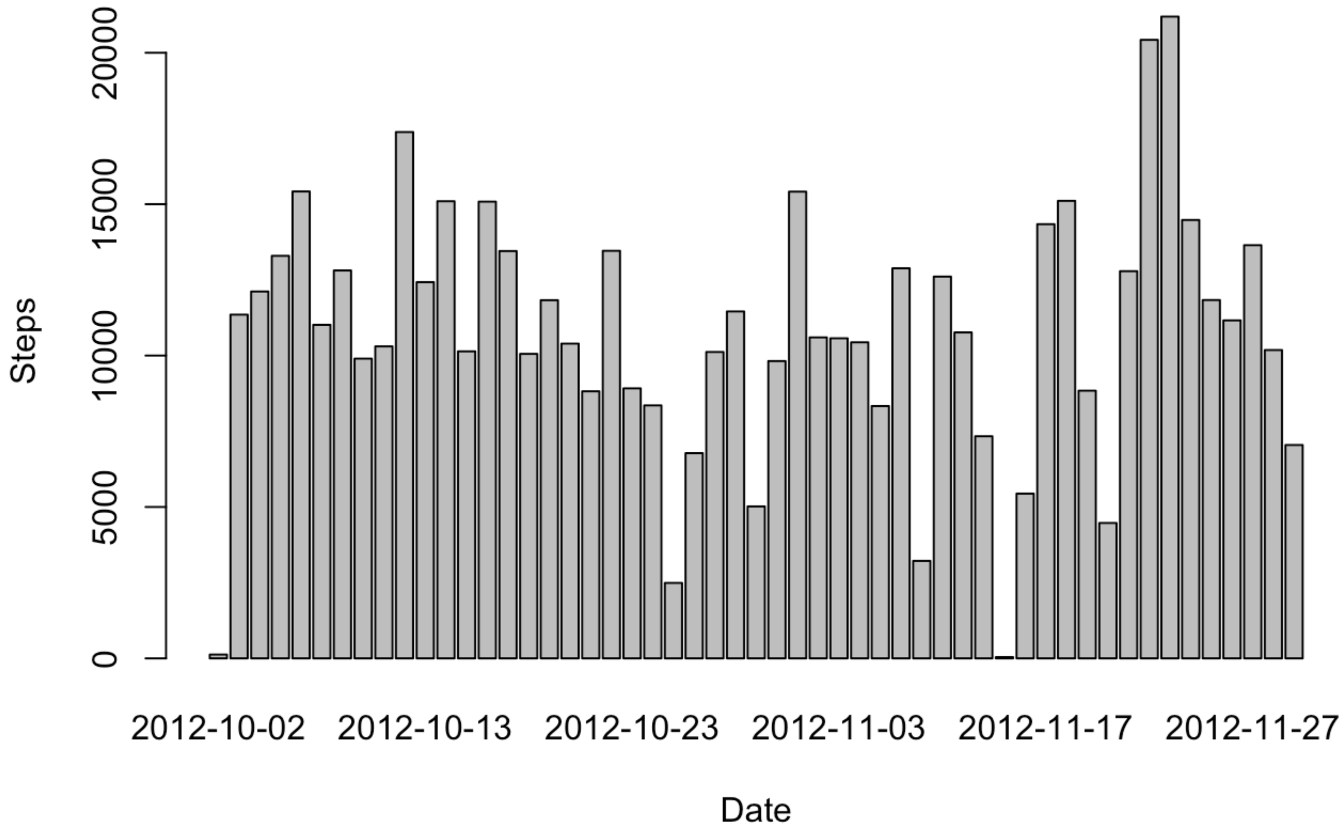
   ```
   We will use the mean of 5 min intervals of steps to populate the missing data values
   ```

3. Creating a new data set and moving the setting the values to the mean of the 5 min time intervals

```
#Copying the data set to new data set
newActivityData <- activityData
meanStepsData<- aggregate(steps ~ interval, data = activityData, FUN = mean, na.rm = T)
for (x in meanStepsData$interval) {
  sapply(x,function (x) newActivityData[ !complete.cases(newActivityData) & (newActivityDat
a$interval == x),1] <- meanStepsData$steps[meanStepsData$interval == x])
}
```

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. 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?

```
new_stepsData<- aggregate(steps ~ date, data = newActivityData, FUN = sum)
barplot(new_stepsData$steps, names.arg = new_stepsData$date, xlab = "Date", ylab = "Steps")
```



Mean

```
print(sprintf("Mean total steps taken per day: %f ", mean(new_stepsData$steps)))
```

```
## [1] "Mean total steps taken per day: 10766.188679 "
```

Median

```
print(sprintf("Median total steps taken per day: %f ", median(new_stepsData$steps)))
```

```
## [1] "Median total steps taken per day: 10765.000000 "
```

Are there differences in activity patterns between weekdays and weekends?

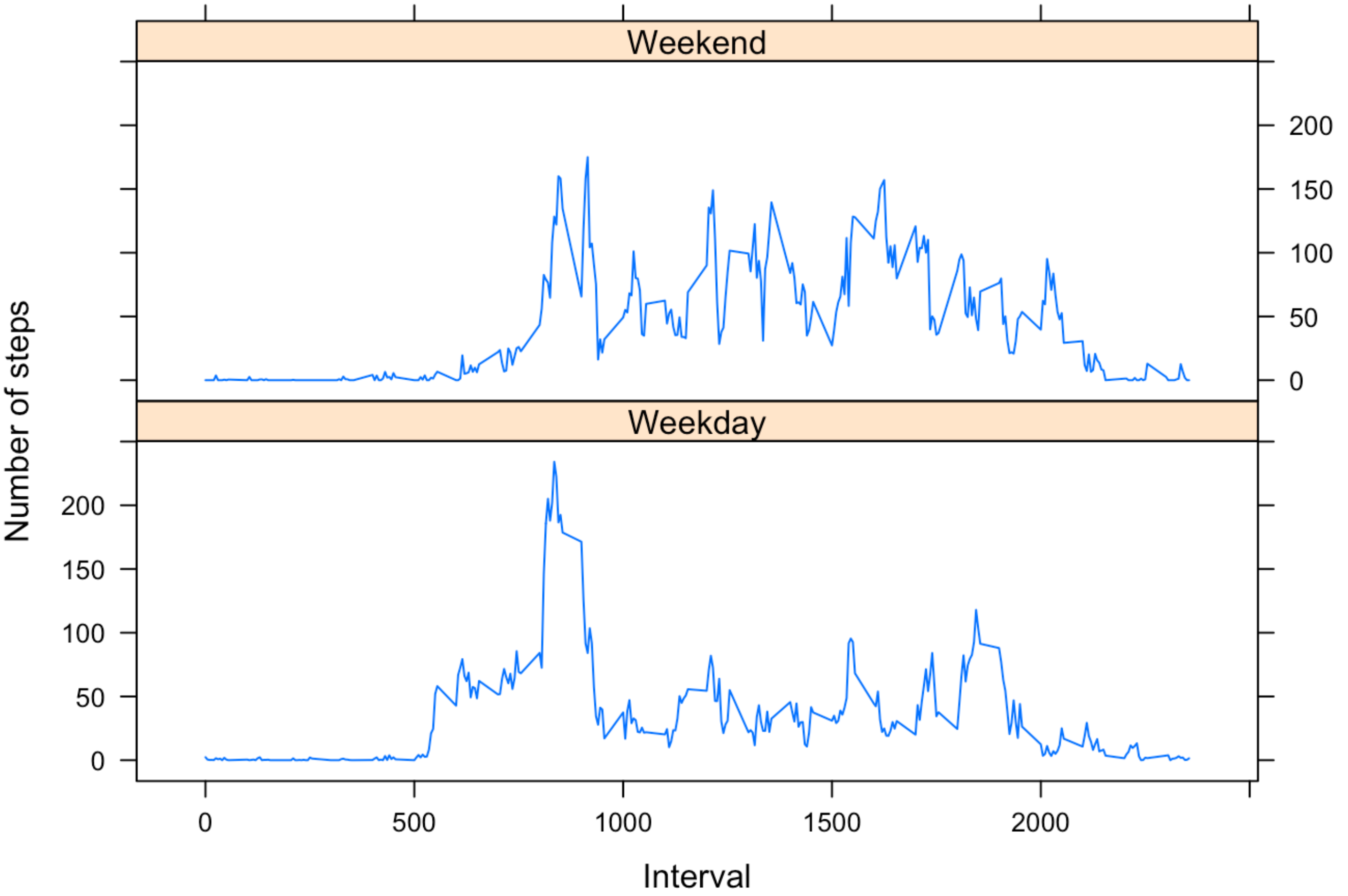
For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

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.

```
dayOfWeek <- function(date) {
    if (weekdays(as.Date(date)) %in% c("Saturday", "Sunday")) {
        day <- "Weekend"
    } else {
        day <- "Weekday"
    }
  }

newActivityData$daytype <- as.factor(sapply(newActivityData$date, dayOfWeek))
```

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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
library("lattice")

dayTypeIntervalSteps <- aggregate(
    data=newActivityData,
    steps ~ daytype + interval,
    FUN=mean
)

xyplot(
    type="l",
    data=dayTypeIntervalSteps,
    steps ~ interval | daytype,
    xlab="Interval",
    ylab="Number of steps",
    layout=c(1,2)
)
```


Weekend
Weekday
Number of steps
Interval
0
50
100
150
200
0
500
1000
1500
2000