project1

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. Assignment

This assignment will be described in multiple parts. You will need to write a report that answers the questions detailed below. Ultimately, you will need to complete the entire assignment in a single R markdown document that can be processed by knitr and be transformed into an HTML file.

Throughout your report make sure you always include the code that you used to generate the output you present. When writing code chunks in the R markdown document, always use echo = TRUE so that someone else will be able to read the code. This assignment will be evaluated via peer assessment so it is essential that your peer evaluators be able to review the code for your analysis.

For the plotting aspects of this assignment, feel free to use any plotting system in R (i.e., base, lattice, ggplot2)

Fork/clone the GitHub repository created for this assignment. You will submit this assignment by pushing your completed files into your forked repository on GitHub. The assignment submission will consist of the URL to your GitHub repository and the SHA-1 commit ID for your repository state.

NOTE: The GitHub repository also contains the dataset for the assignment so you do not have to download the data separately. Loading and preprocessing the data

Show any code that is needed to

```
Load the data (i.e. read.csv())

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

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.

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

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

What is the average daily activity pattern?

```
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)

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

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

```
Calculate and report the total number of missing values in the dataset (i.e. the total number of rows wi

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.

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?
```

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.

```
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.

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
The plot should look something like the following, which was creating using simulated data:
```

Your plot will look different from the one above because you will be using the activity monitor data. Note that the above plot was made using the lattice system but you can make the same version of the plot using any plotting system you choose.

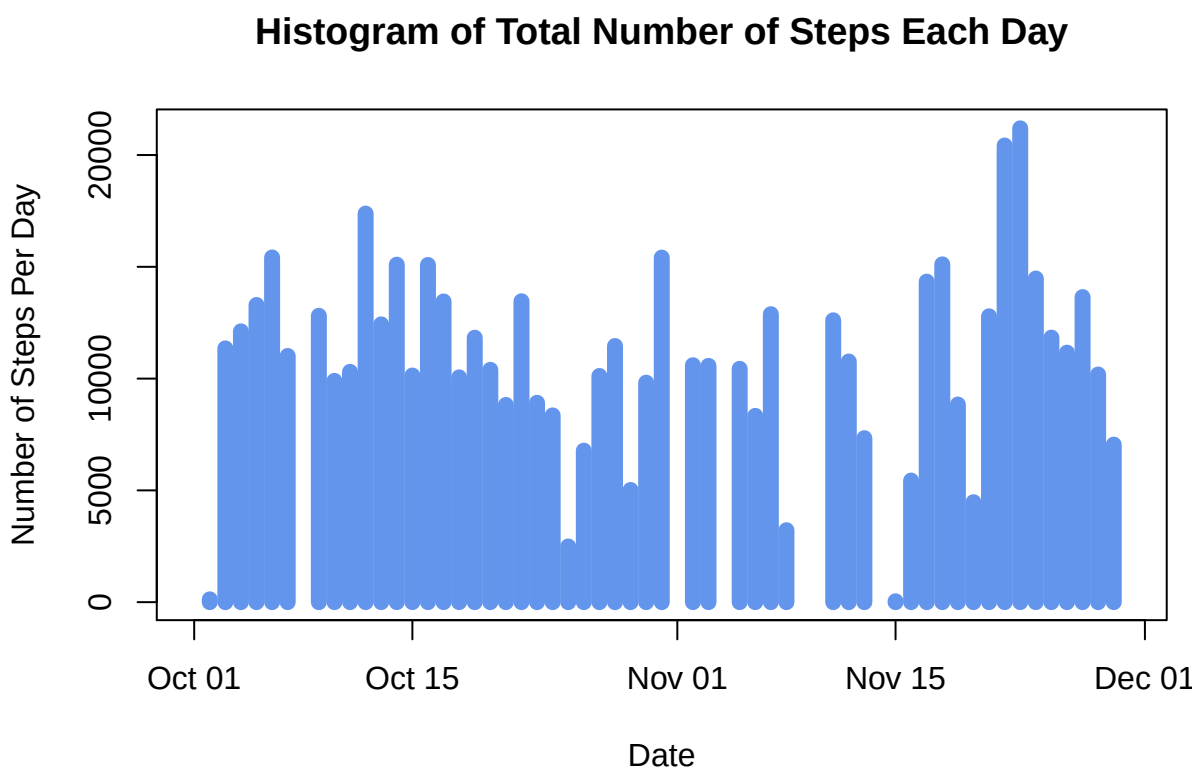
Beginning of the solution

Loading and preprocessing the data

```
raw <- read.csv("./activity.csv")
raw$date <- as.Date(raw$date)
# using str() to have a look at data, the date format needs to be changed
```

Histogram, mean and median values calculation

```
library(plyr)
daysum <- ddply(raw, .(date), function(x) sum(x$steps))
with(daysum, plot(x=date, y=V1, xlab = "Date", ylab="Number of Steps Per Day", main="Histogram of Total
```




```
paste("Mean value of total number of steps taken per day: ",as.integer(mean(daysum$V1, na.rm=T)))
```

```
## [1] "Mean value of total number of steps taken per day:  10766"
```

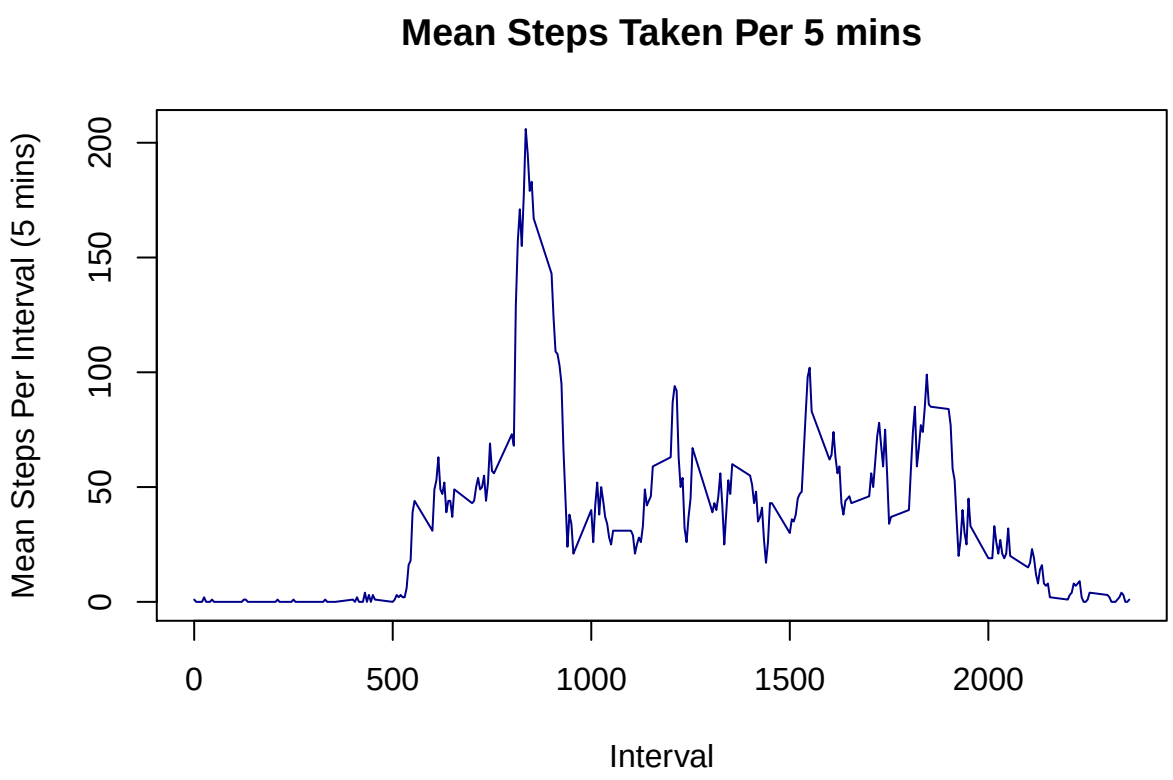
```
paste("Median value of total number of steps taken per day: ",median(daysum$V1, na.rm=T))
```

```
## [1] "Median value of total number of steps taken per day:  10765"
```

Strategy for imputing missing values: calculate the mean values for each interval, and replace the NA values with these mean values

```
library(ggplot2)

interval_mean <- ddply(raw, .(interval), function(x) mean(x$steps, na.rm=TRUE))

# convert the mean values to integer since it represents numbers of steps.
```

```
interval_mean$V1 <- as.integer(interval_mean$V1)
names(interval_mean)[2] <- "mean.step"

# draw a time series plot of the average number of steps taken versus the 5-minute intervals

with(interval_mean,plot(interval, mean.step, type="l", xlab="Interval", ylab="Mean Steps Per Interval (
```

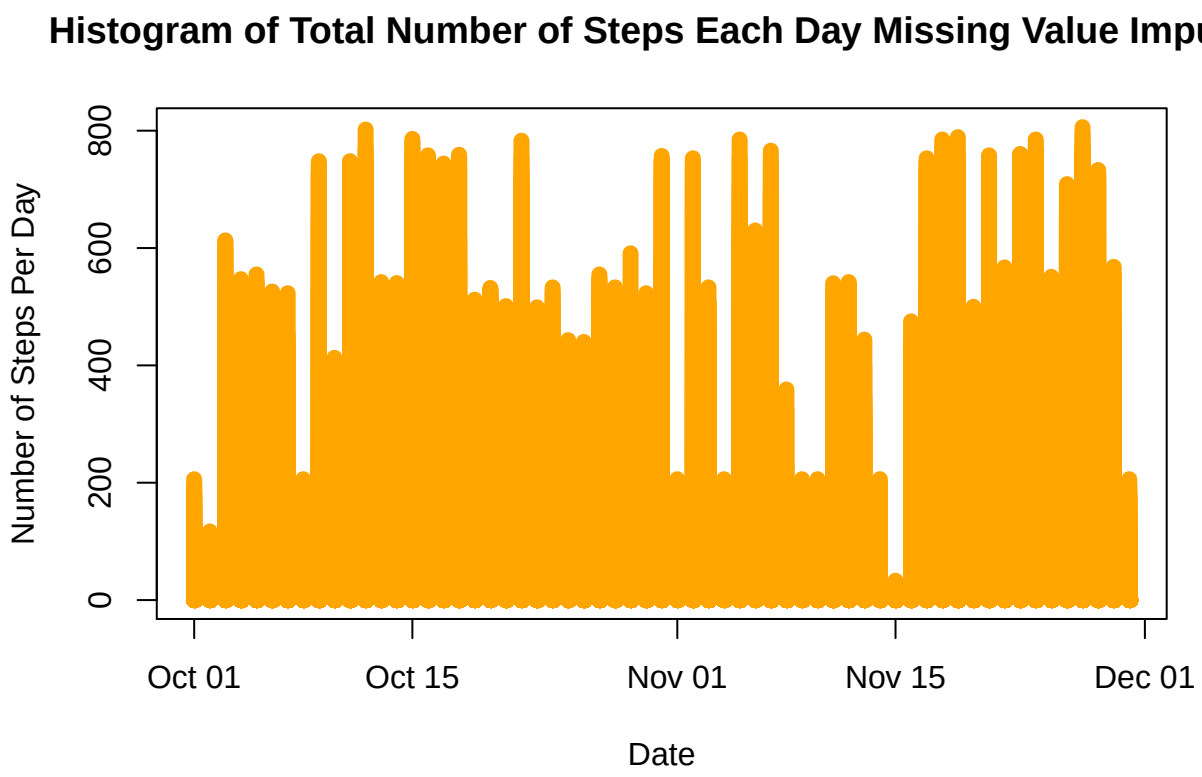



```
# maxium number of steps taken per interval(on average)

paste("Maximum steps taken in a 5-min interval", max(interval_mean$mean.step))
```

```
## [1] "Maximum steps taken in a 5-min interval 206"
```

```
# locate missing values

match <- raw[is.na(raw$steps), ]

# subset the raw data to get those missing steps values

match <- merge(match, interval_mean, by="interval", all=TRUE)
imputedata <- raw
imputedata <- merge(imputedata, match, by=c("date", "interval"), all=TRUE)

# merge data set with imputed values

imputedata$steps.new <- rowSums(imputedata[,3:5], na.rm=TRUE)
```

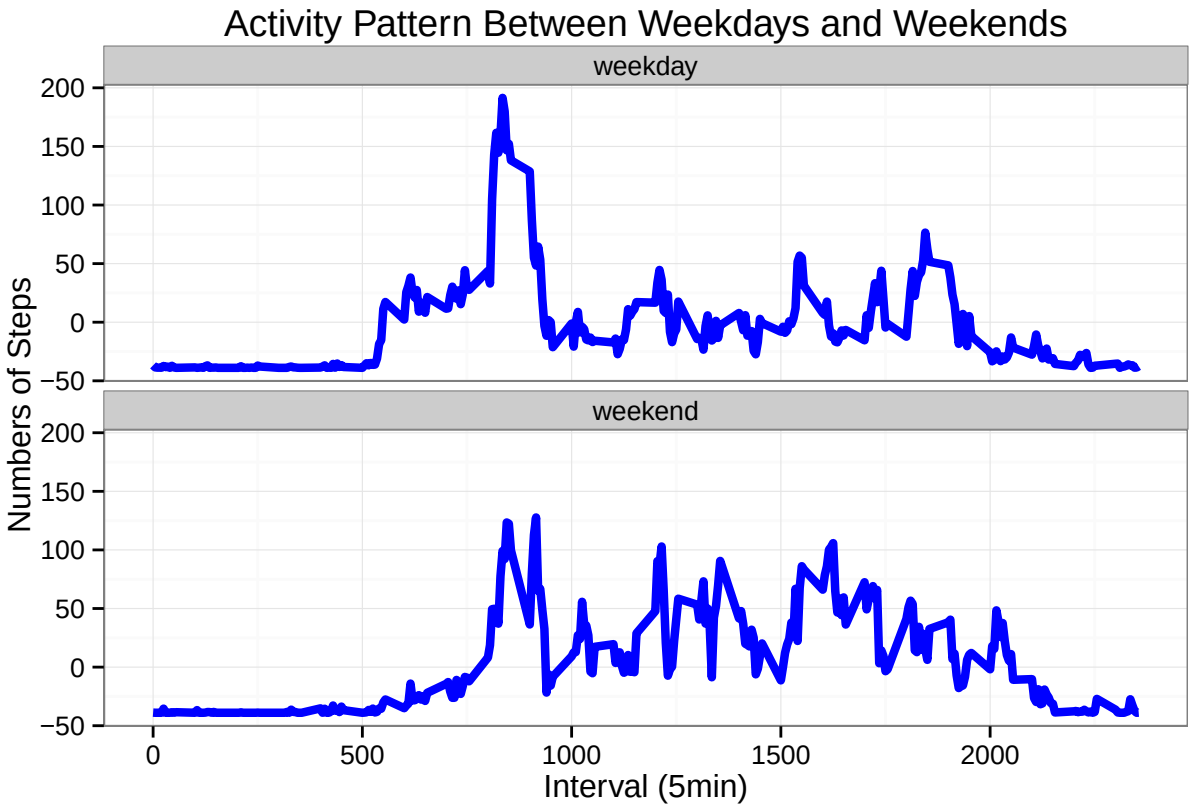
```
# sum the values to remove NA values and store the new steps values in a new column
# next, remove unnecessary column and make the date set clean

imputedata <- imputedata[,c(1,2,6)]

# plot the histogram

with(imputedata, plot(x=date, y=steps.new, xlab = "Date", ylab="Number of Steps Per Day",
                      main="Histogram of Total Number of Steps Each Day\ Missing Value Imputed",
                      type="h", col="orange", lwd =8))
```



Next, we are going to add a type column to the imputedata set to distinguish weekdays and weekends

```
imputedata$type <- ifelse(weekdays(imputedata$date) %in% c("Saturday", "Sunday"),
                          "weekend", "weekday")
imputedata$type <-as.factor(imputedata$type)

# calculated total numbers of steps for weekdays and weekends separately and save
# is in a new table called simulated

simulated <- ddply(imputedata, .(interval,type), summarise, mean=mean(steps.new))

# plot the patterns

ggplot(data=simulated,
       aes(x=interval, y=mean-mean(simulated$mean)))+
    geom_line(color = "blue", lwd=1.5)+facet_wrap(~type,ncol=1)+
    theme(strip.background=element_rect("cornflowerblue"))+
    xlab("Interval (5min)")+ylab("Numbers of Steps")+
```

```
    ggtitle("Activity Pattern Between Weekdays and Weekends")+
    theme_bw()
```



This is the end of project 1 for Reproducible Research course.