DataMonitor

Kalyan

April 6, 2018

Read and display the dataset into R

Note: the dataset is present in the same directory as the R Markdown file.

```
data <- read.csv("activity.csv")
#Preprocess Dates to Date type
data$date <- as.Date(data$date)
head(data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

Plot the histogram of total steps taken each day:

```
#Sum all the steps grouping by day
library(dplyr)
```

```
## 
## Attaching package: 'dplyr'
```

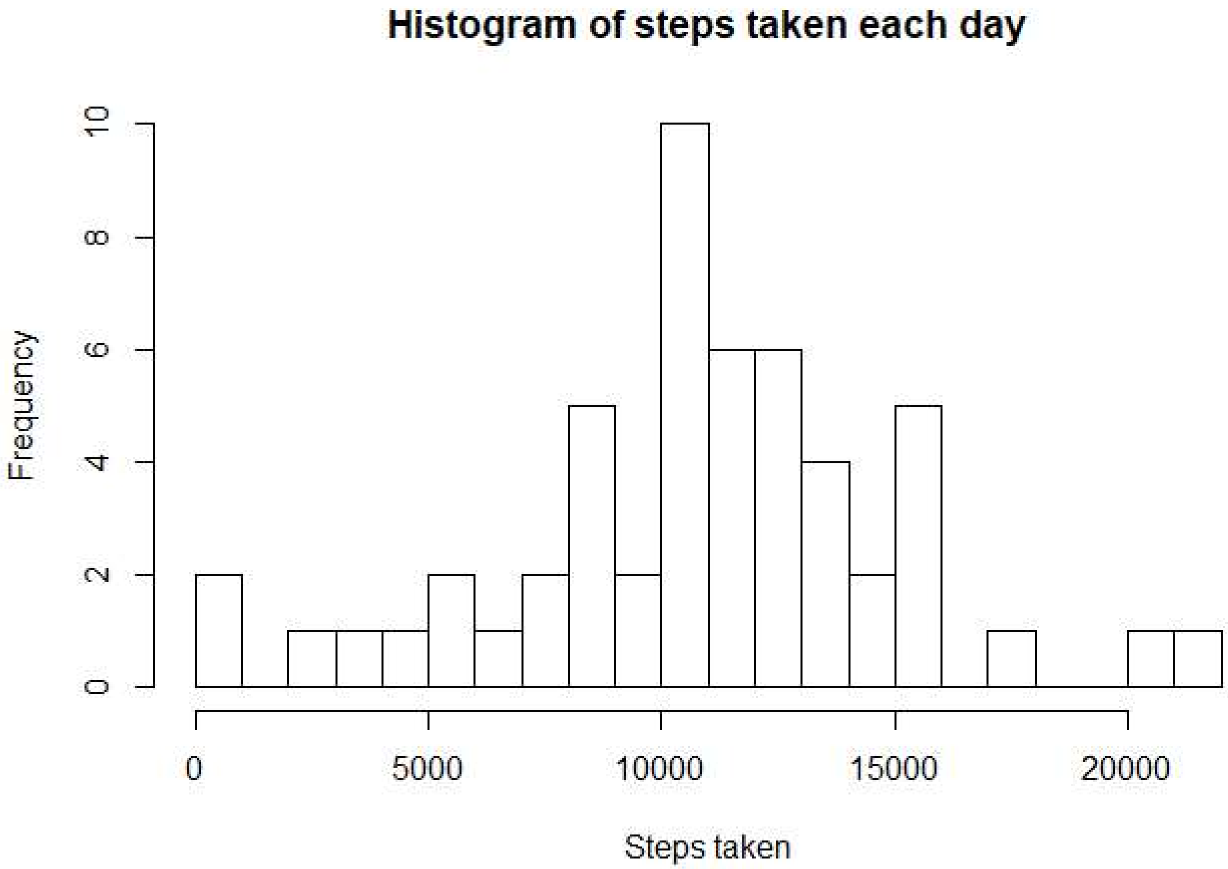
```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

```
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```
steps <- data %>% group_by(date) %>% summarize(tot_steps = sum(steps))

#remove NAs
steps <- steps[!is.na(steps$tot_steps), ]

#plot the histogram
hist(steps$tot_steps, breaks = 30, main = "Histogram of steps taken each day", xlab = "Steps tak
en")
```



Calculate mean and median of steps for each day

```
#calculate mean of steps
mean_steps <- mean(steps$tot_steps)

#calculate median of steps
median_steps <- median(steps$tot_steps)

#print them
mean_steps
```

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

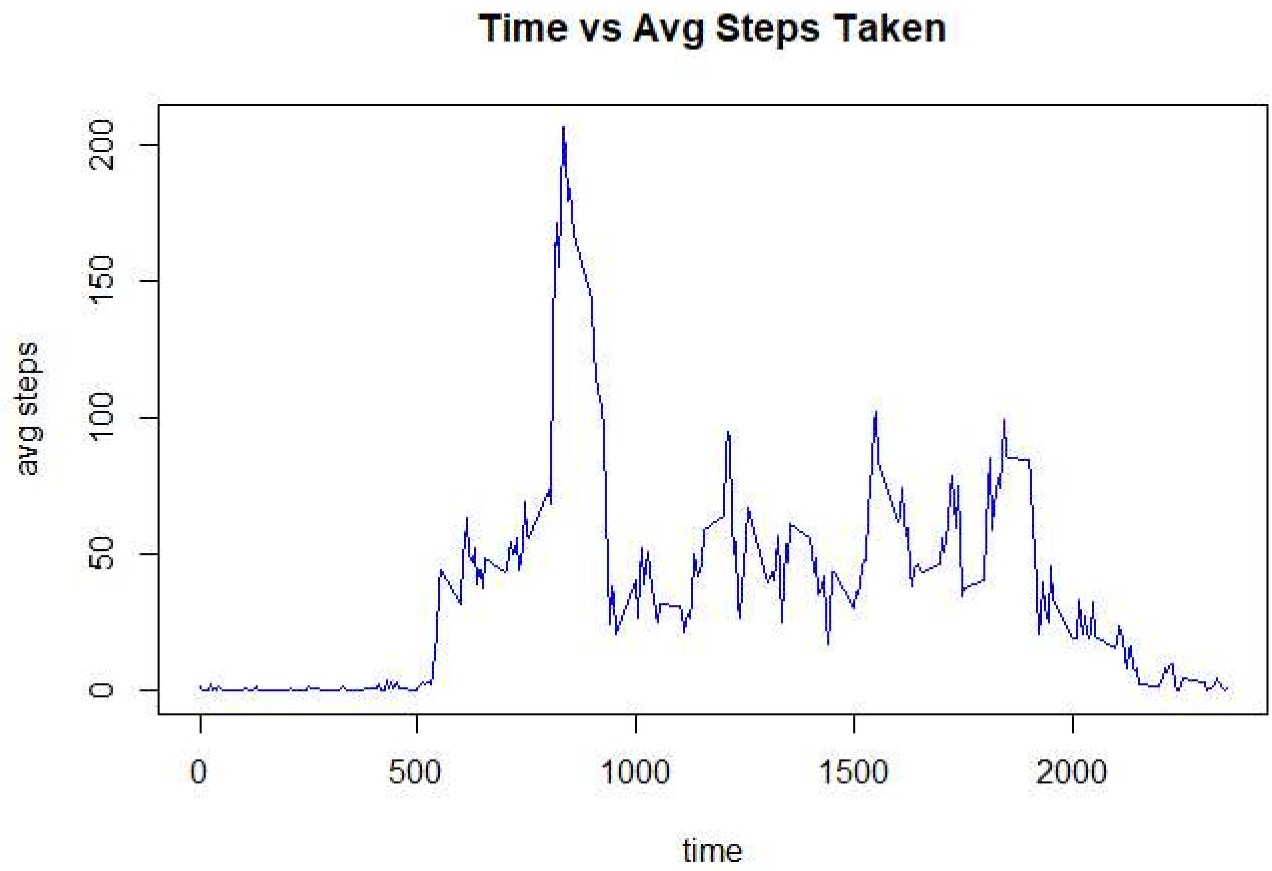
```
median_steps
```

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

Plot time series of the average number of steps taken

```
avg_steps <- data %>% na.omit() %>% group_by(interval) %>% summarize(avg_s = mean(steps))

plot(avg_steps, type = "l", col = "blue", xlab = "time", ylab = "avg steps", main = "Time vs Avg
 Steps Taken")
```



The 5 minute interval that on average contains max steps

```
#Find the interval number that contains max number of average steps.
as.numeric(avg_steps[avg_steps$avg_s == max(avg_steps$avg_s),"interval"])
```

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

Impute missing values

Number of NAs:

```
naSteps <- is.na(data$steps)
sum(naSteps)
```

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

Number of NAs after imputing:

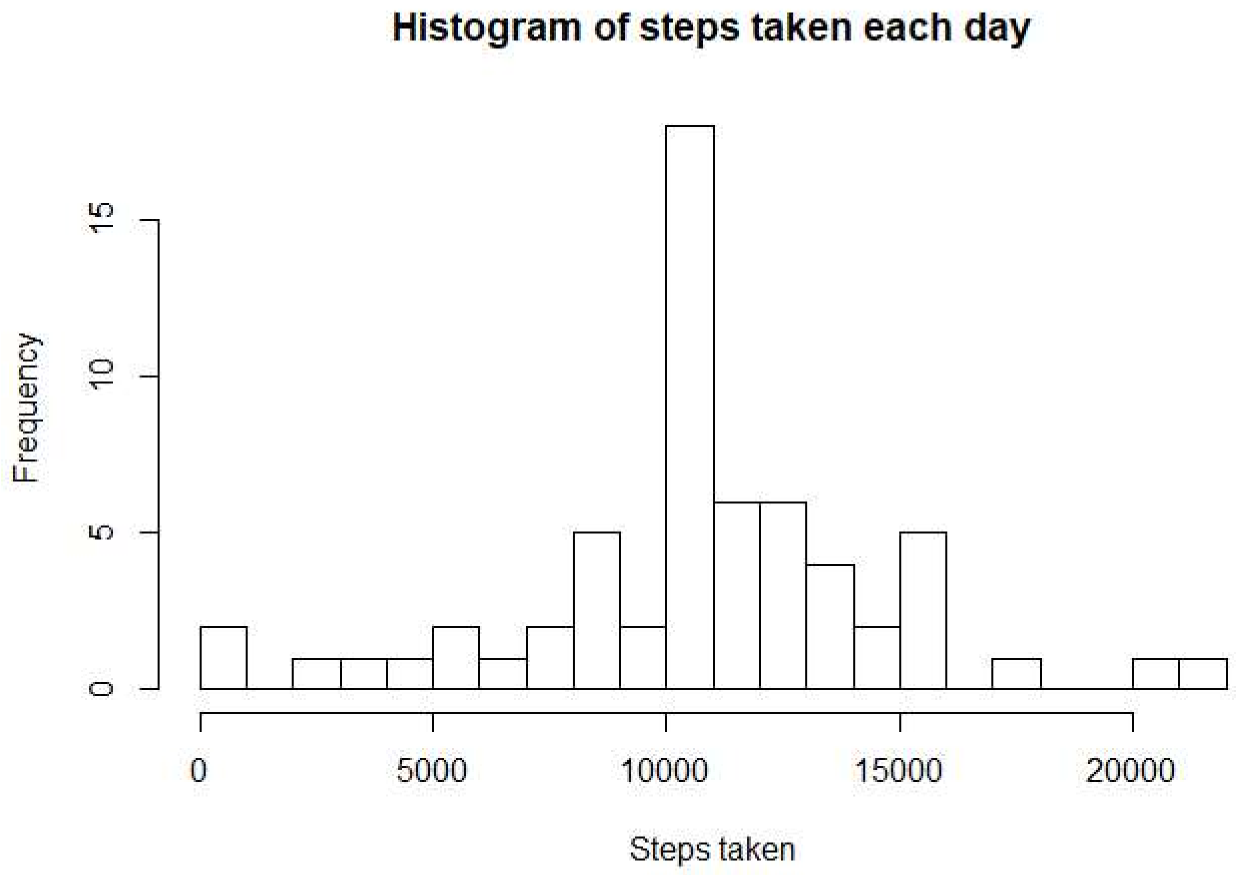
```
#Replacing NAs with the mean for that interval
data$imputed_data <- replace(data$steps, naSteps, avg_steps$avg_s)
imputed_steps <- data %>% group_by(date) %>% summarize(tot_steps = sum(imputed_data))

#Number of NAs after imputing
sum(is.na(data$imputed_data))
```

```
## [1] 0
```

Histogram of steps:

```
hist(imputed_steps$tot_steps, breaks = 30, main = "Histogram of steps taken each day", xlab = "Steps taken")
```



Mean and Median post imputing

```
#calculate mean of steps
mean_steps <- mean(imputed_steps$tot_steps)

#calculate median of steps
median_steps <- median(imputed_steps$tot_steps)

#print them
mean_steps
```

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

```
median_steps
```

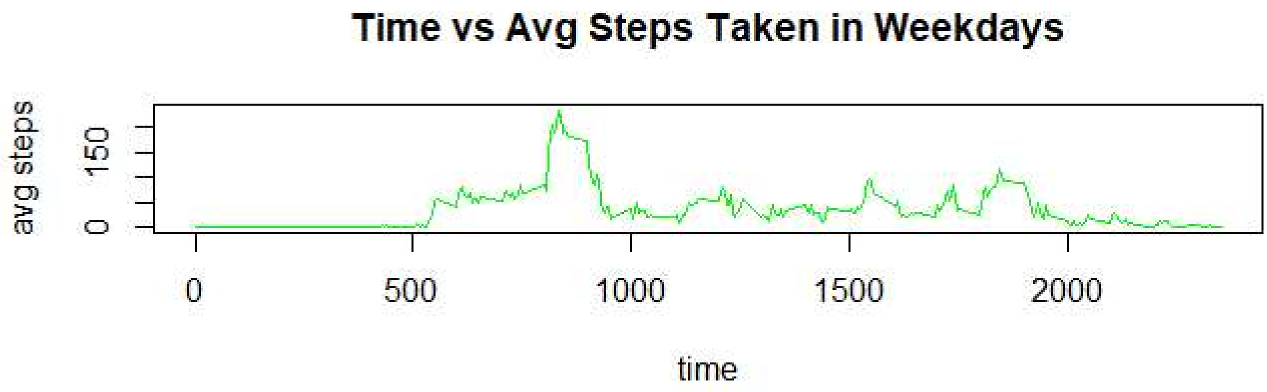
```
## [1] 10766.19
```

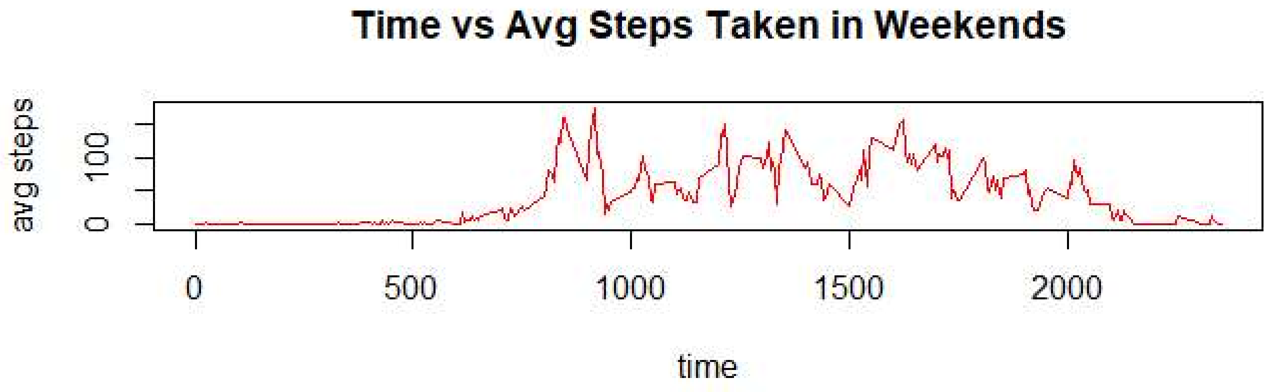
```
data$dayname <- weekdays(data$date)
data$daytype <- ifelse(data$dayname=="Saturday" | data$dayname=="Sunday", "weekend",
                           "weekday" )

avg_steps_wkday <- data[data$daytype == "weekday",] %>% na.omit() %>% group_by(interval) %>% sum
marize(avg_s = mean(steps))

avg_steps_wkend <- data[data$daytype == "weekend",] %>% na.omit() %>% group_by(interval) %>% sum
marize(avg_s = mean(steps))


par(mfrow = c(2,1))
plot(avg_steps_wkday, type = "l", col = "green", xlab = "time", ylab = "avg steps", main = "Time
 vs Avg Steps Taken in Weekdays")
plot(avg_steps_wkend, type = "l", col = "red", xlab = "time", ylab = "avg steps", main = "Time v
s Avg Steps Taken in Weekends")
```


Time vs Avg Steps Taken in Weekdays
avg steps
time


Time vs Avg Steps Taken in Weekends
avg steps
time