

PA1__template

Mike McGunagle

August 17, 2016

1. Code for reading in the dataset and/or processing the data

```
setwd("C:/Users/Mike/Desktop/R_programming_2/Reproducible_research")

activity<-read.csv("activity.csv")
head(activity)
```

```
##      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
```

```
str(activity)
```

```
## 'data.frame':   17568 obs. of  3 variables:
## $ steps      : int  NA NA NA NA NA NA NA NA NA NA ...
## $ date       : Factor w/ 61 levels "2012-10-01","2012-10-02",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ interval: int   0 5 10 15 20 25 30 35 40 45 ...
```

```
#convert date to a date format
```

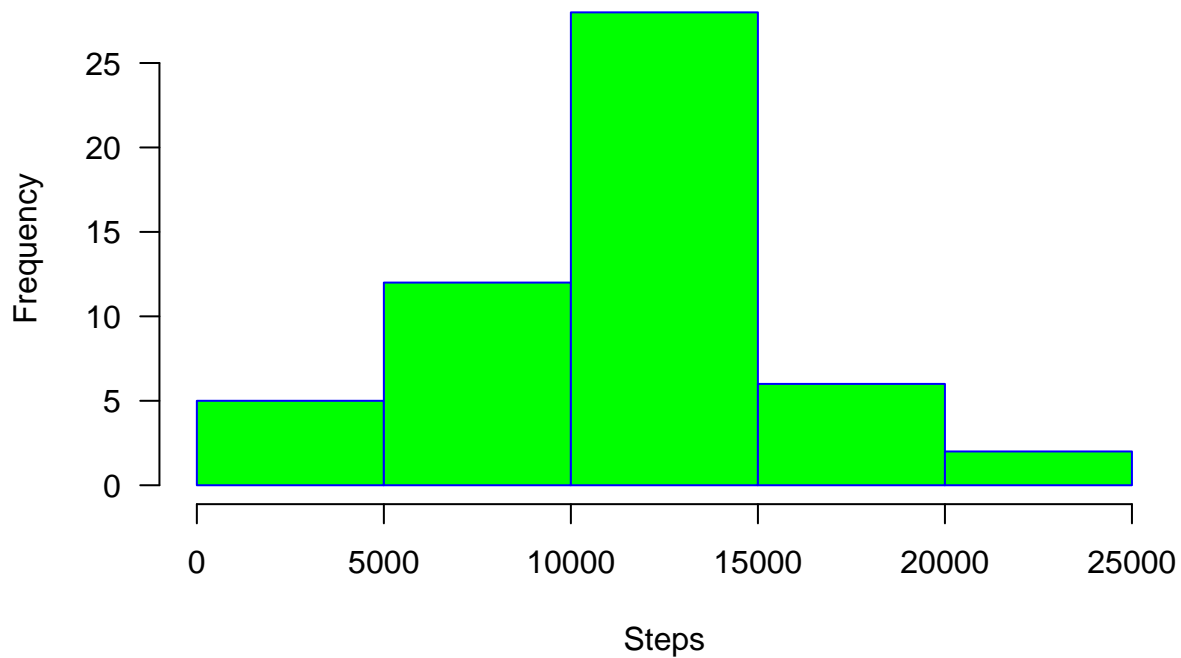
```
activity$date<-as.Date(activity$date, as.character('%Y-%m-%d'))
```

2. Histogram of the total number of steps taken each day

```
library(plyr)
daily_steps<-ddply(activity,~date,summarise,sum=sum(steps))
```

```
hist(daily_steps$sum,      main="Histogram for Daily Steps",      xlab="Steps",      border="blue",      col="red")
```

Histogram for Daily Steps



3. Mean and median number of steps taken each day

```
mean_daily_steps<-mean(daily_steps$sum, na.rm = TRUE)
mean_daily_steps
```

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

```
median_daily_steps<-median(daily_steps$sum, na.rm = TRUE)
median_daily_steps
```

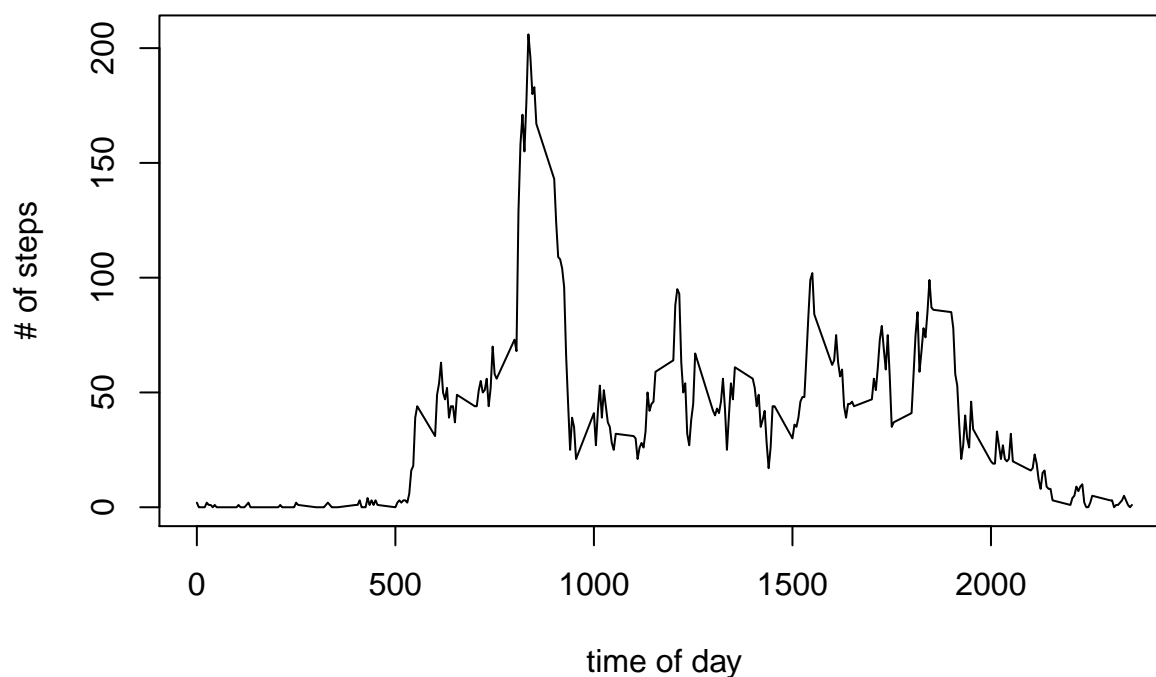
```
## [1] 10765
```

4. Time series plot of the average number of steps taken

```
interval_steps<-ddply(activity,~interval,summarise,step_mean=mean(steps, na.rm = TRUE))
interval_steps$step_mean<-round(interval_steps$step_mean, 0)
```

```
plot(interval_steps$interval, interval_steps$step_mean, type = 'l', main = "Average steps per interval")
```

Average steps per interval



5. The 5-minute interval that, on average, contains the maximum number of steps

```
max_interval<-max(interval_steps$step_mean)
subset(interval_steps, step_mean == max_interval)
```

```
##      interval step_mean
## 104      835      206
```

6. Code to describe and show a strategy for imputing missing data

```
# number of missing data
sum(is.na(activity))
```

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

```
# fill in missing values
head(interval_steps)
```

```
##      interval step_mean
## 1          0          2
## 2          5          0
## 3         10          0
## 4         15          0
## 5         20          0
## 6         25          2
```

```

merge_steps<-merge(activity, interval_steps, by ="interval")
merge_steps<-merge_steps[order(merge_steps$date),]

missing<-subset(merge_steps, is.na(steps))
missing$steps<-0
non_missing<-subset(merge_steps, steps>=0)

for (i in 1:2304){
  missing[i,2]<-missing[i,4]
}

new_data<-rbind(missing, non_missing)
new_data<-new_data[order(new_data$date),]

# re-calculated steps each data with missing data filled in as 5 minute
# interval average
new_daily_steps<-ddply(new_data,~date,summarise,sum=sum(steps))

#mean
mean(new_daily_steps$sum)

```

```
## [1] 10765.64
```

```
median(new_daily_steps$sum)
```

```
## [1] 10762
```

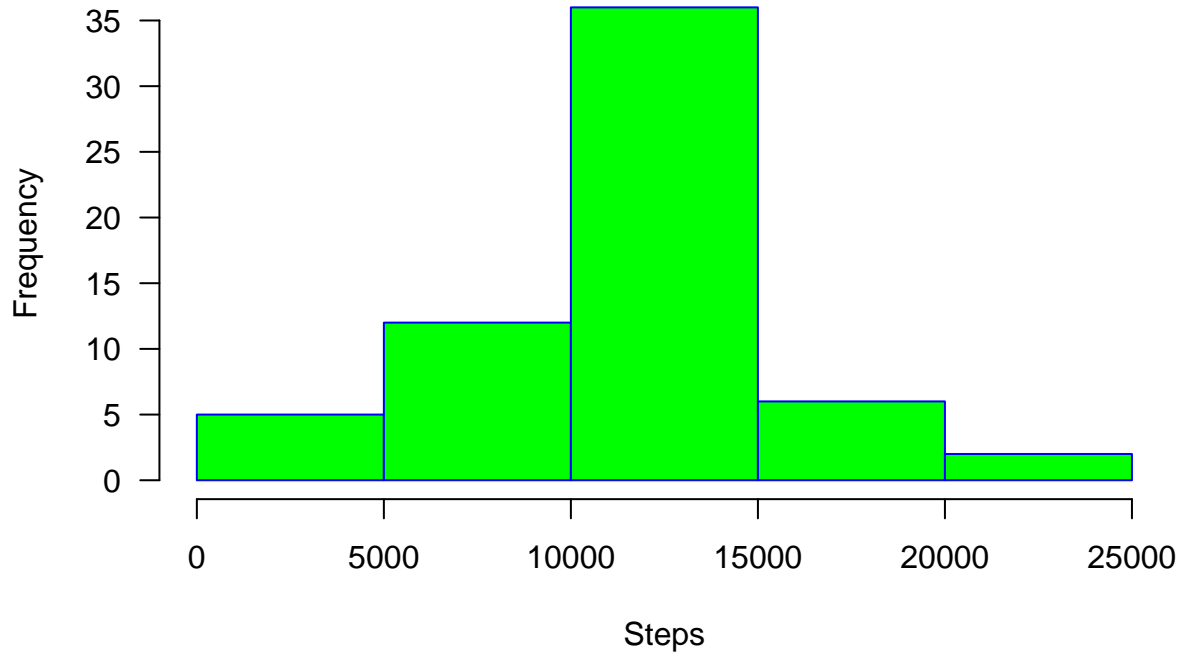
7. Histogram of the total number of steps taken each day after missing values are imputed

```

#histogram
hist(new_daily_steps$sum,      main="Histogram for Daily Steps with missing data filled in",      xlab="S

```

Histogram for Daily Steps with missing data filled in



8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
##  
## The following object is masked from 'package:plyr':  
##  
##     here
```

```
weekdays1 <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')  
new_data$wDay <- factor((weekdays(new_daily_steps$date) %in% weekdays1),  
  levels=c(FALSE, TRUE), labels=c('weekend', 'weekday'))  
  
weekday_steps<-subset(new_data, wDay == "weekday")  
weekend_steps<-subset(new_data, wDay == "weekend")  
  
wday_int_steps<-ddply(weekday_steps,~interval,summarise,step_mean=mean(steps, na.rm = TRUE))  
wday_int_steps$step_mean<-round(wday_int_steps$step_mean, 0)  
  
wend_int_steps<-ddply(weekend_steps,~interval,summarise,step_mean=mean(steps, na.rm = TRUE))  
wend_int_steps$step_mean<-round(wend_int_steps$step_mean, 0)
```

```

par(mfrow=c(2,1))
plot(wday_int_steps$interval, wday_int_steps$step, type = 'l', main = "Average steps per interval - Week Day")
plot(wend_int_steps$interval, wend_int_steps$step, type = 'l', main = "Average steps per interval - Week End")

```

