Reproducible Research: Peer Assessment 1

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.

(See full description at https://github.com/rdpeng/RepData_PeerAssessment1)

Required packages:

```r
library(data.table)
library(dplyr)
library(ggplot2)
library(lubridate)
```

Loading and preprocessing the data

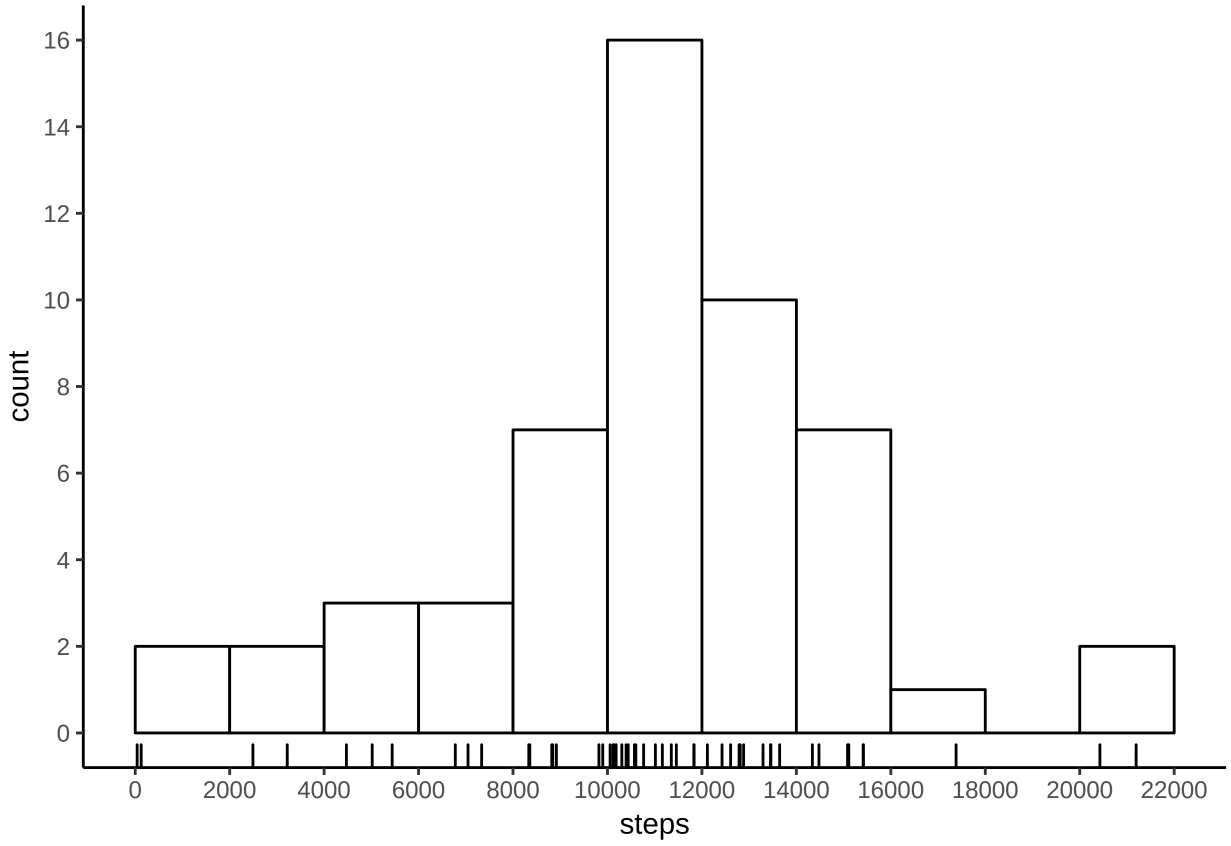
```r
file_name <- "activity.csv"
if(!file.exists(file_name)){
        temp <- tempfile()
        download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", temp)
        unzip(temp)
        unlink(temp)
}

activity <- fread(file_name)
activity <- activity %>%
        mutate(date=ymd(date), day_of_week=weekdays(date),
               weekdayend=ifelse(day_of_week %in% c("Saturday", "Sunday"), "weekend", "weekday"))
```

```
## Warning in as.POSIXlt.POSIXct(x, tz): unknown timezone 'zone/tz/2018c.1.0/
## zoneinfo/Australia/Melbourne'
```

What is the mean total number of steps taken per day?

```r
daily_steps <- activity %>%
        filter(!is.na(steps)) %>%
        group_by(date) %>%
        summarise(steps=sum(steps))
```

```r
ggplot(daily_steps, aes(x=steps)) +
        geom_histogram(binwidth=2000, boundary=0, fill="white", col="black") +
        geom_rug() +
        scale_x_continuous(breaks=seq(0, 22000, 2000)) +
        scale_y_continuous(breaks=seq(0, 16, 2)) +
        theme_classic()
```



The mean number of steps per day is

```r
mean(daily_steps$steps)
```

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

The median number of steps per day is

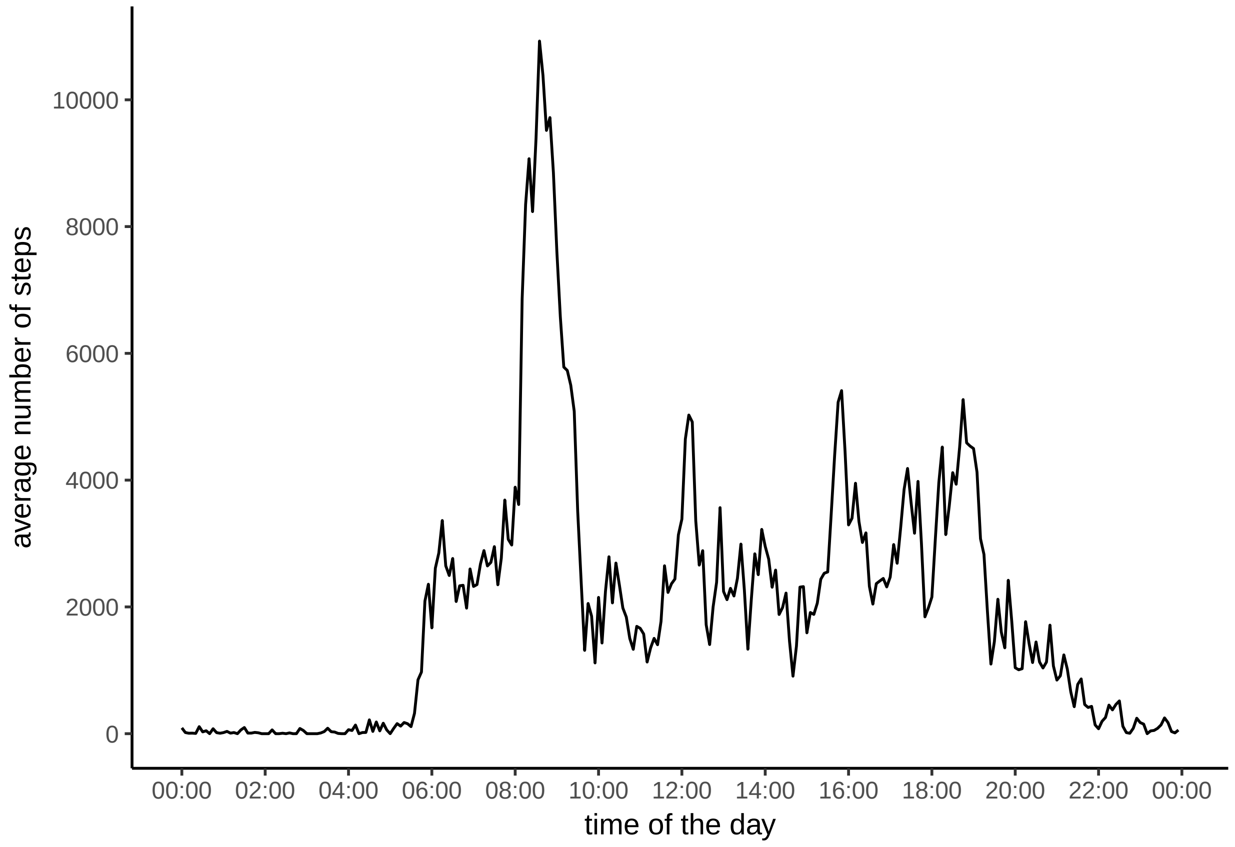
```r
median(daily_steps$steps)
```

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

The person on average hits the healthy goal of 10000 steps a day.

What is the average daily activity pattern?

```r
time_seq <- format(seq.POSIXt(as.POSIXct(Sys.Date()),
                              as.POSIXct(Sys.Date()+1), by= "2 hours"), "%H:%M", tz="GMT")

daily_average <- activity %>%
        filter(!is.na(steps)) %>%
        group_by(interval) %>%
        summarise(steps=sum(steps))

ggplot(daily_average, aes(x=seq(0,287,1), y = steps)) +
        geom_line() +
        scale_x_continuous(labels=time_seq, breaks=seq(0,288, length=length(time_seq))) +
        scale_y_continuous(breaks=seq(0, 10000, 2000)) +
        xlab("time of the day") +
        ylab("average number of steps") +
        theme_classic()
```



Most of the walking happens between 8 and 10am. On average across all the days in the dataset, the maximum number of steps is attained at 8:35

```
filter(daily_average, steps==max(steps))$interval
```

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

Imputing missing values

The total number of missing values in the dataset is

```
sum(is.na(activity$steps))
```

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

Now let's see which days have missing values:

```
activity %>% filter(is.na(steps)) %>% group_by(date, day_of_week) %>% summarise(n=n())
```

```
## # A tibble: 8 x 3
## # Groups:   date [?]
##         date day_of_week     n
##       <date>       <chr> <int>
## 1 2012-10-01      Monday   288
## 2 2012-10-08      Monday   288
## 3 2012-11-01    Thursday   288
## 4 2012-11-04      Sunday   288
## 5 2012-11-09      Friday   288
## 6 2012-11-10    Saturday   288
## 7 2012-11-14   Wednesday   288
## 8 2012-11-30      Friday   288
```

n=288 for all days, which means if a number of steps for some interval is missing, then it's missing for all intervals in that day. The value should be inferred from other days.

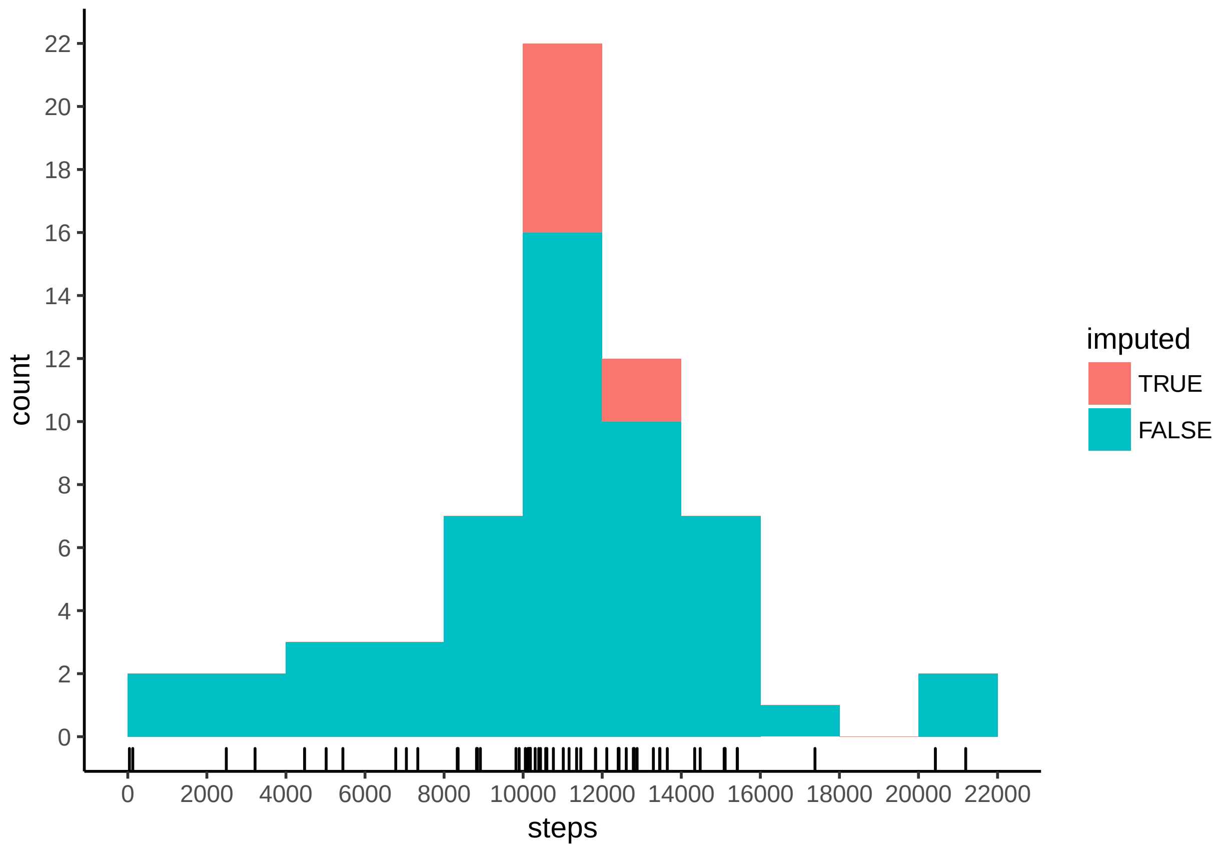
The missing value can be imputed as mean/median for that interval based on all days, or the days can be split by `day_of_week` variable or weekend/weekday characteristic and then compute mean/median for each group.

Probably later on should try all methods and compare the results, but for now here is the imputation of missing values with mean value for that interval in weekend/weekday groups.

```
mean_steps <- activity %>%
        filter(!is.na(steps)) %>%
        group_by(interval, weekdayend) %>%
        summarise(steps=mean(steps))

imputed_activity <- activity
for (i in 1:nrow(imputed_activity)){
        if (is.na(imputed_activity$steps[i])) {
                # this is slow, ~2min, do log search in the future with data.table ordered indexes?
                val <- filter(mean_steps, weekdayend==imputed_activity$weekdayend[i],
                              interval==imputed_activity$interval[i])$steps
                imputed_activity$steps[i] <- val
        }
}
```

Total number of steps per day based on the imputed data set.

```
daily_steps_imputed <- imputed_activity %>%
        mutate(imputed=is.na(activity$steps), imputed=factor(imputed, levels=c(TRUE, FALSE))) %>%
        group_by(date, imputed) %>%
        summarise(steps=sum(steps))

ggplot(daily_steps_imputed, aes(x=steps, fill=imputed)) +
        geom_histogram(binwidth=2000, boundary=0) +
        geom_rug() +
        scale_x_continuous(breaks=seq(0, 22000, 2000)) +
        scale_y_continuous(breaks=seq(0, 22, 2)) +
        theme_classic()
```



The new mean is

```r
mean(daily_steps_imputed$steps)
```

```
## [1] 10762.05
```

The new median is

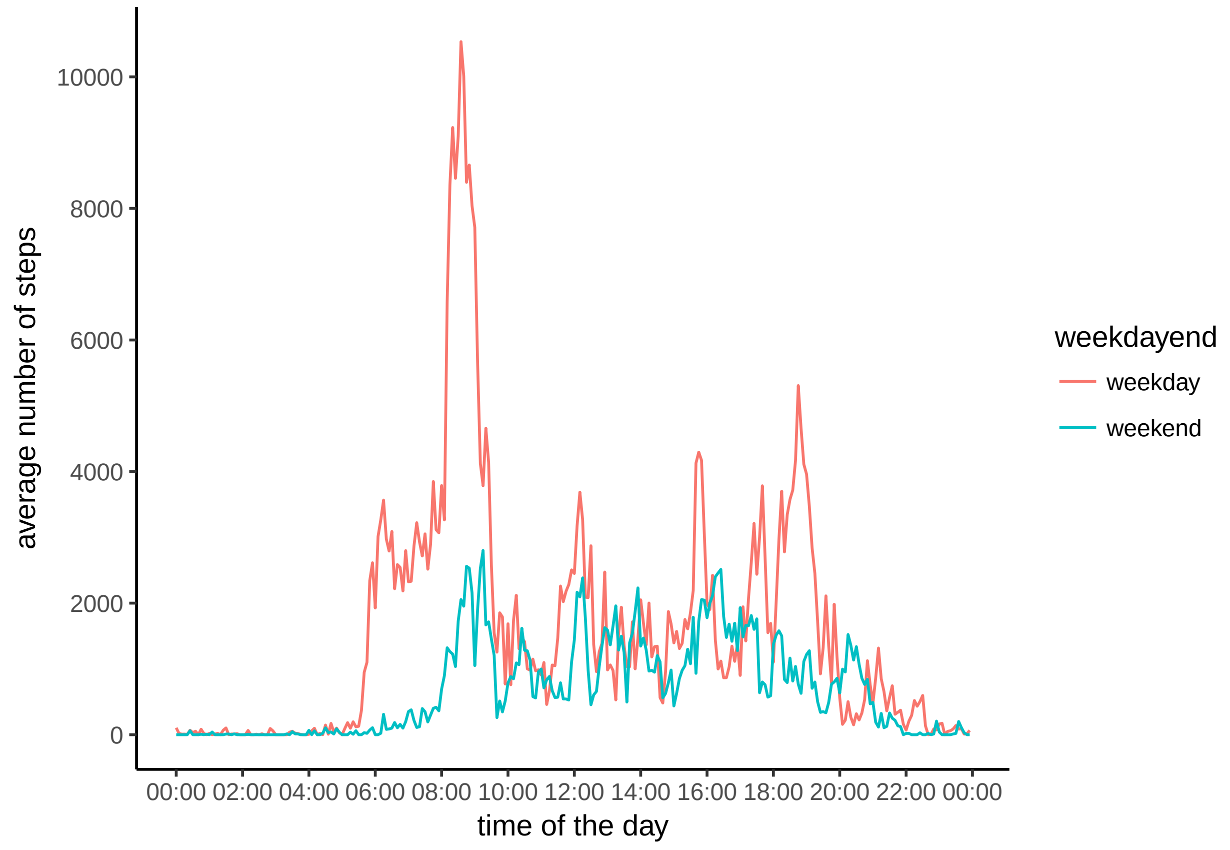
```r
median(daily_steps_imputed$steps)
```

```
## [1] 10571
```

The mean is almost unchanged (great!), the median is slightly lower (still awesome!), imputation has almost no effect on mean and median.

Any differences between weekdays and weekends?

```r
daily_average_imputed <- imputed_activity %>%
        group_by(interval, weekdayend) %>%
        summarise(steps=sum(steps))

ggplot(daily_average_imputed, aes(x=rep(seq(0,287,1), each=2),
                                  y=steps, col=weekdayend, group=weekdayend)) +
        geom_line() +
        scale_x_continuous(labels=time_seq, breaks=seq(0,288, length=length(time_seq))) +
        scale_y_continuous(breaks=seq(0, 10000, 2000)) +
        xlab("time of the day") +
        ylab("average number of steps") +
        theme_classic()
```



As expected, weekends are less active than weekdays, especially in the morning 6-10am and evening 6-8pm