Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

The data is contained in a csv file, with three columns: the number of steps, the date (‘YYYY-MM-DD’ format) and an id label for each 5 minute interval. Missing step values where coded as NA.

Code that is needed to: 1. Load the data (i.e. read.csv())

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

```
setwd("/home/alfredo/Documentos/Coursera/RepData_PeerAssessment1")
nike_data <- read.csv( unzip("activity.zip"),
                sep=",",
                na.strings = "NA",
                colClasses =c("numeric","Date","numeric"))
str(nike_data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: num  0 5 10 15 20 25 30 35 40 45 ...
```

```
head(nike_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
```

```
tail(nike_data)
```

```
##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

```
library(ggplot2)
library(plyr)
library(lubridate)
```

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

Attaching package: 'lubridate'

The following object is masked from 'package:plyr':

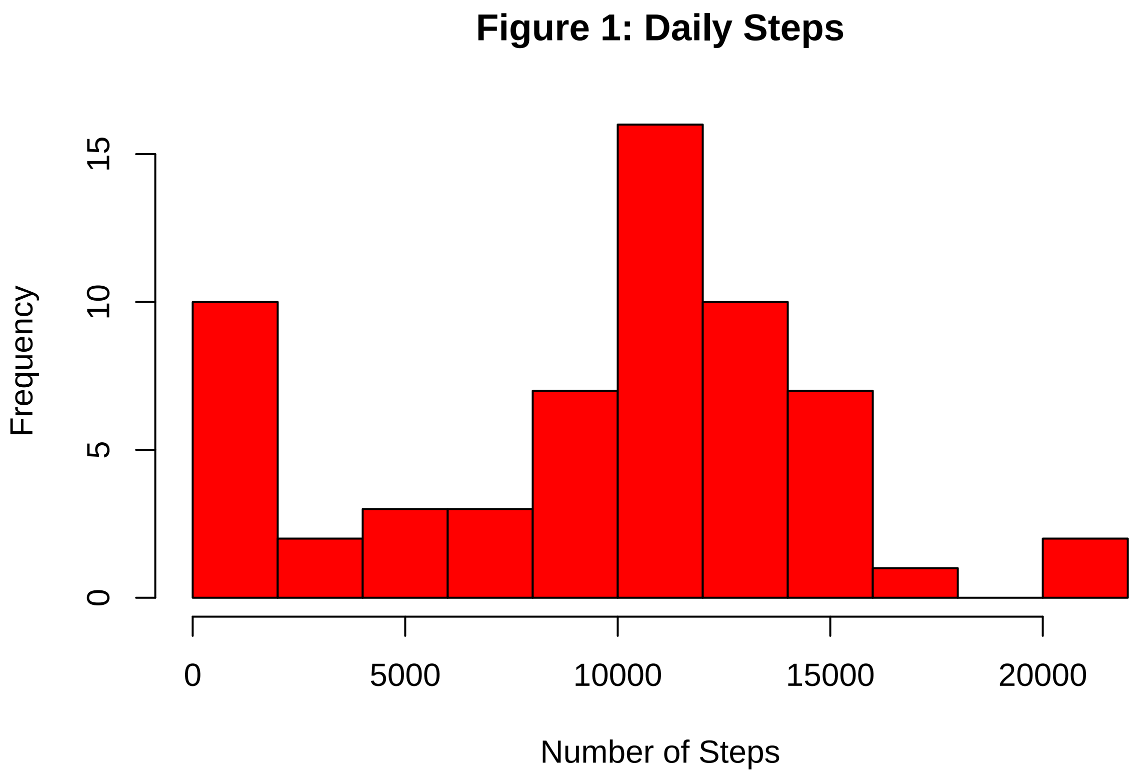
here

```
library(lattice)
library(knitr)
```

What is mean total number of steps taken per day?

For this part of the assignment, the missing values in the dataset are ignored. 1. Make a histogram of the total number of steps taken each day 2. Calculate and report the mean and median total number of steps taken per day

```
step_day <- tapply(nike_data$steps,nike_data$date,function(x) sum(x,na.rm=TRUE))
hist(step_day, breaks = 15, col="red",xlab="Number of Steps", main="Figure 1: Daily Steps")
```



```r
#Mean total number of steps taken per day:
step_mean <-mean(step_day, na.rm = T)
step_mean
```

```
## [1] 9354.23
```

```r
summary(step_day)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0    6778   10400    9354   12810   21190
```

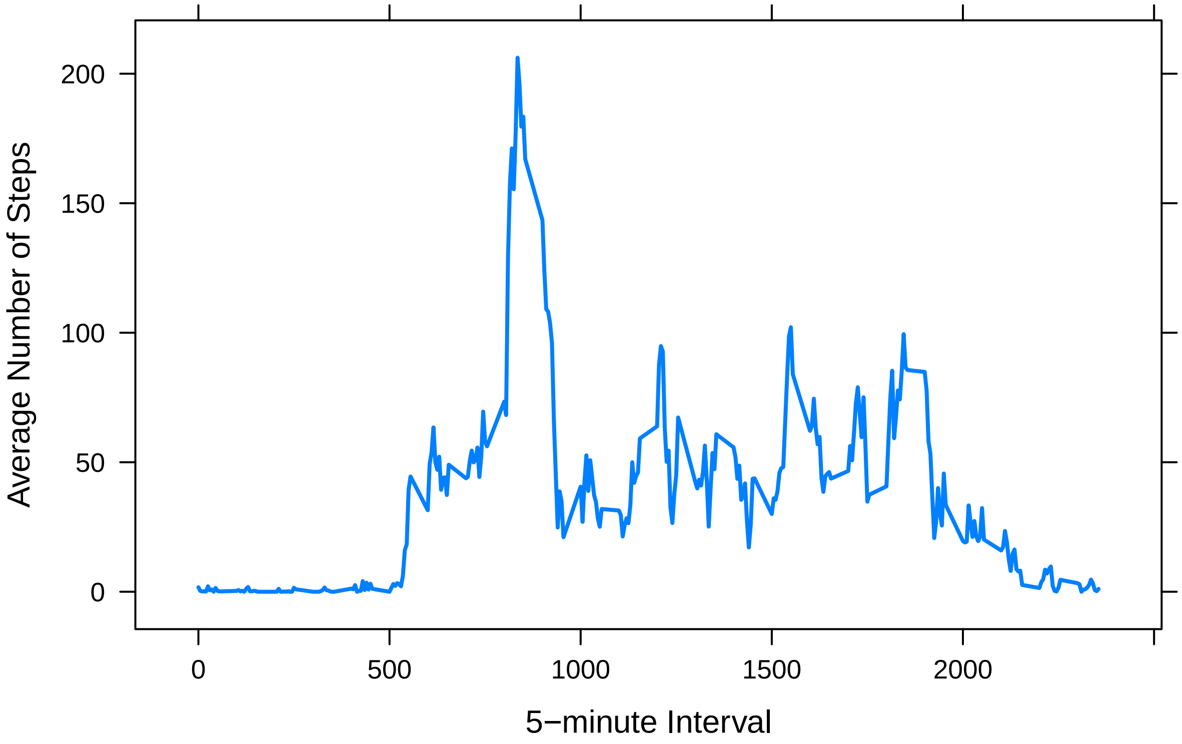
What is the average daily activity pattern?

```r
steps_pattern <- aggregate(nike_data$steps ~ nike_data$interval, nike_data, FUN=mean, na.rm=T)

names(steps_pattern) <- c("interval","average_steps")

xyplot(steps_pattern$average_steps ~ steps_pattern$interval,
       type = "l", ylab = "Average Number of Steps",
       xlab ="5-minute Interval",
       main = "Figure 2a: Time Series Plot", lwd = 2)
```

Figure 2a: Time Series Plot



```
max_steps <- which.max(steps_pattern$average_steps)
max_steps
```

```
## [1] 104
```

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

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

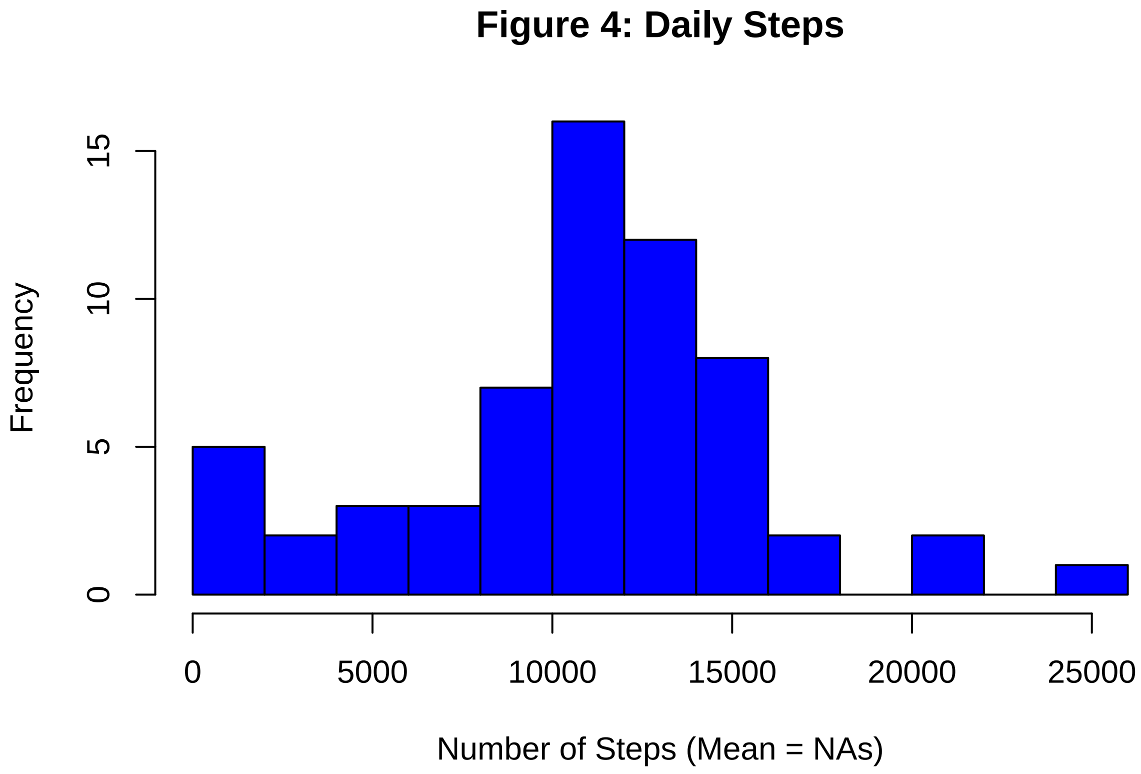
```
## [1] 2304
```

```
sub_nas <- nike_data[is.na(nike_data),]
sub_nas$steps <- merge(steps_pattern, sub_nas)$average_steps

##Create a new dataset that is equal to the original dataset but with the missing data filled in.

nike_data_fill <- nike_data
nike_data_fill[is.na(nike_data),] <- sub_nas
daily_steps_fill <- tapply(nike_data_fill$steps,nike_data_fill$date,function(x) sum(x,na.rm=TRUE))

##Make a histogram of the total number of steps taken each day and Calculate and report the mean and mec
hist(daily_steps_fill, breaks = 15, col="blue",xlab="Number of Steps (Mean = NAs)", main="Figure 4: Dail
```

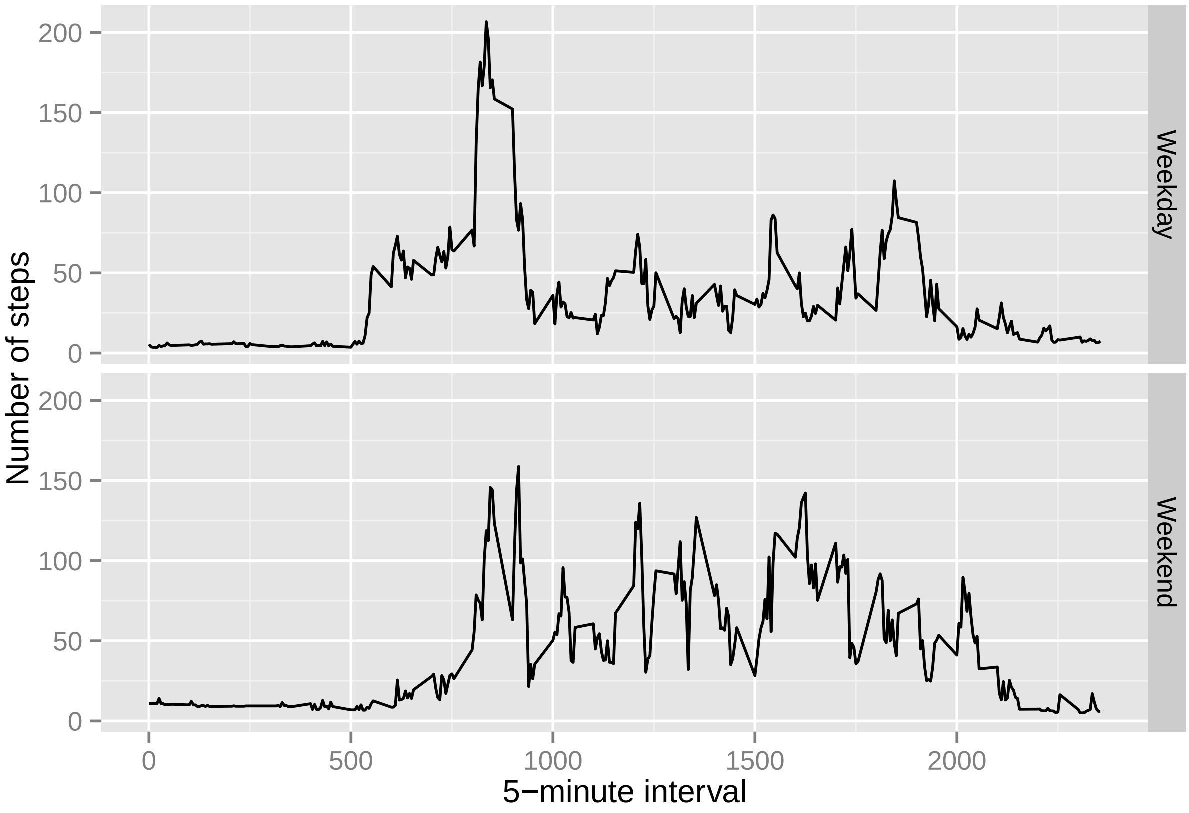


Are there differences in activity patterns between weekdays and weekends?

The dataset with the filled-in missing values is used. 1. A new factor variable is created in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day. 2. 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 is constructed, averaged across all weekday days or weekend days (y-axis).

```r
daytype <- function(date) {
        if (weekdays(as.Date(date)) %in% c("Saturday", "Sunday")) {
                "Weekend"
        } else {
                "Weekday"
        }
}
nike_data_fill$daytype <- as.factor(sapply(nike_data_fill$date, daytype))
nike_data_fill$day <- sapply(nike_data_fill$date, FUN = daytype)

averages <- aggregate(steps ~ interval + day, data = nike_data_fill, mean)
ggplot(averages, aes(interval, steps)) + geom_line() + facet_grid(day ~ .) +
    xlab("5-minute interval") + ylab("Number of steps")
```


Number of steps
Weekday
Weekend
5−minute interval