

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 research 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 coursera web site:

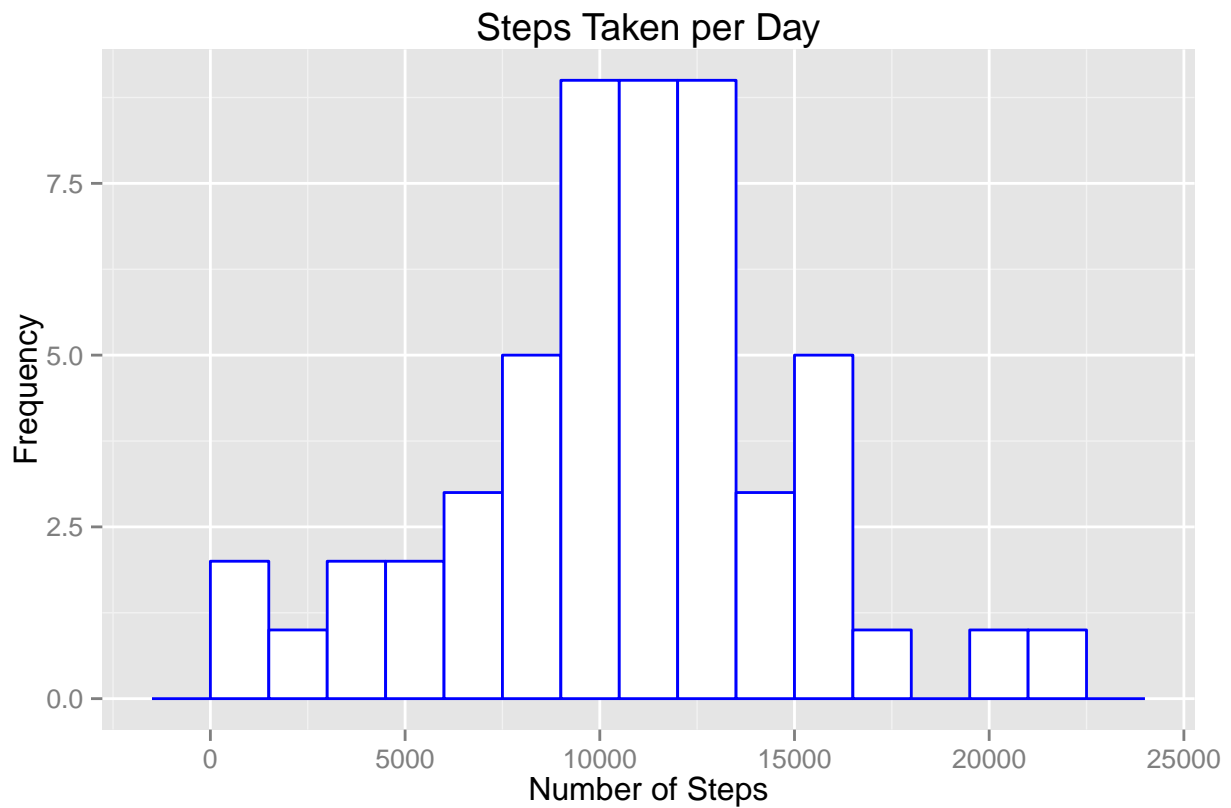
*Dataset: [Activity monitoring data](#) [52K]

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset. At first Load data file into the Rstudio Environment:

```
## Loading required libraries
library(ggplot2)
library(plyr)
## Loading and preprocessing the data
data <- read.csv("activity.csv", colClasses = c("numeric", "Date", "numeric"))
```

Plot total number of steps per a day

```
## What is mean total number of steps taken per day?
byDay <- aggregate(steps ~ date, data, sum)
byDay <- cbind(byDay, label = rep("with.na", nrow(byDay)))
ggplot(byDay, aes(x = steps)) + geom_histogram(binwidth = 1500, colour = "blue",
  fill = "white") + labs(title = "Steps Taken per Day", x = "Number of Steps",
  y = "Frequency")
```



Mean and median number of steps

Calculate and report the mean and median total number of steps

```
mean(byDay$steps)
```

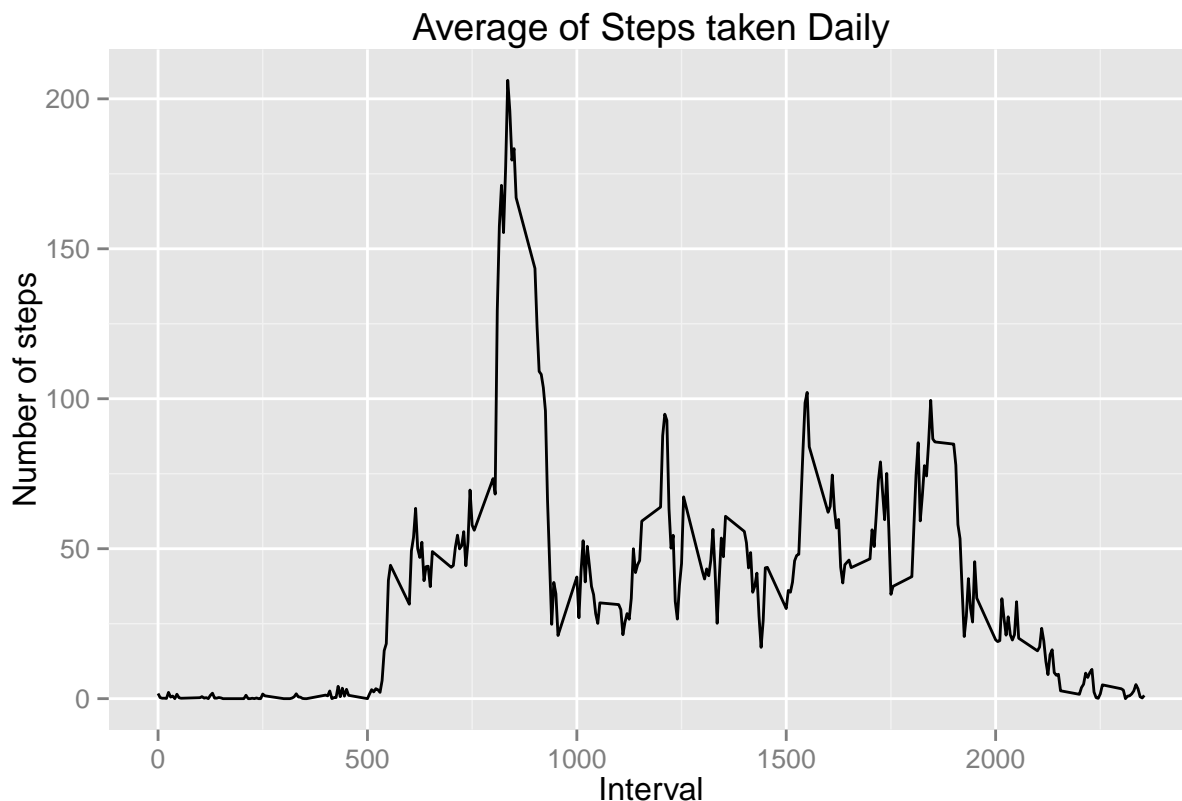
```
## [1] 10766.19
```

```
median(byDay$steps)
```

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

Average daily activity pattern

```
## What is the average daily activity pattern?
byInterval <- aggregate(steps ~ interval, data, mean, na.rm = TRUE)
ggplot(byInterval, aes(x = interval, y = steps)) + geom_line() + labs(title = "Average of Steps taken D",
  x = "Interval", y = "Number of steps")
```



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

```
interval <- aggregate(steps ~ interval, data = data, FUN = mean)
interval$interval[which.max(interval$steps)]
```

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

Imputing missing values

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

```
## Imputing missing values
sum(is.na(data))
```

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

```
data.impute <- adply(data, 1, function(x) if (is.na(x$steps)) {
  x$steps = round(byInterval[byInterval$interval == x$interval, 2])
  x
} else {
  x
})
```

Differences between weekdays and weekends

```
## Are there differences in activity patterns between weekdays and weekends?  
data.weekend <- subset(data.impute, weekdays(date) %in% c("Saturday", "Sunday"))  
data.weekday <- subset(data.impute, !weekdays(date) %in% c("Saturday", "Sunday"))
```

Average steps per interval for each dataset

```
# Obtain the average steps per interval for each dataset  
data.weekend <- aggregate(steps ~ interval, data.weekend, mean)  
data.weekday <- aggregate(steps ~ interval, data.weekday, mean)  
  
# By plotting we add a label  
data.weekend <- cbind(data.weekend, day = rep("weekend"))  
data.weekday <- cbind(data.weekday, day = rep("weekday"))
```

Combine the subsets and a specify the levels

```
# Combine the subsets and a specify the levels  
data.week <- rbind(data.weekend, data.weekday)  
levels(data.week$day) <- c("Weekend", "Weekday")  
  
ggplot(data.week, aes(x = interval, y = steps)) + geom_line() + facet_grid(day ~  
  .) + labs(x = "Interval", y = "Number of steps")
```

