Reproducible Research Project 1

Magia Kotsits

February 27, 2016

Load data

```r
# Load activity data
data <- read.csv("./activity.csv")
summary(data)
```

```
##      steps                date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5  
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2  
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0  
##  NA's   :2304     (Other)   :15840
```

```r
str(data)
```

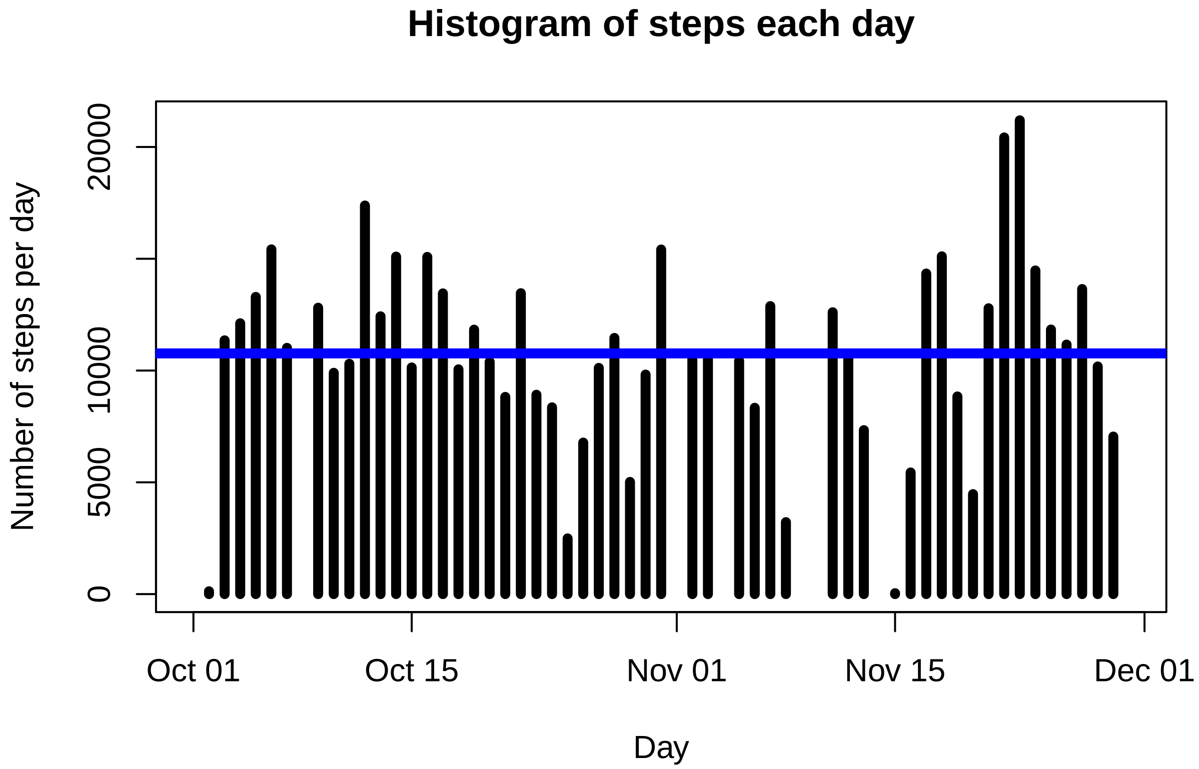
```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Preprocess data and prepare it for plotting

```r
# Convert variable 'date'from factor to date
data$date <- as.Date(data$date)
# Load reshape2 library to use melt and dcast and ggplot for plotting
library(reshape2)
library(ggplot2)
# Melt data frame to prepare for casting by date:
# We set id variable to 'date'and measure variable to steps.
# We get a table with multiple values for steps taken each day.
#We ignore the missing values in the dataset.
data_melt <- melt(data, id.vars="date", measure.vars="steps", na.rm=FALSE)
# Cast data frame to find number of steps per day
data_cast <- dcast(data_melt, date ~ variable, sum)
```

Histogram of the total number of steps taken each day

```
# Plot histogram with frequency of steps by day. Blue line shows the mean value
plot(data_cast$date, data_cast$steps, type="h", main="Histogram of steps each day", xlab="Day", ylab="Nu
abline(h=mean(data_cast$steps, na.rm=TRUE), col="blue", lwd=5)
```



What is mean total number of steps taken per day?

```
paste("Mean steps/day =", mean(data_cast$steps, na.rm=TRUE))
```

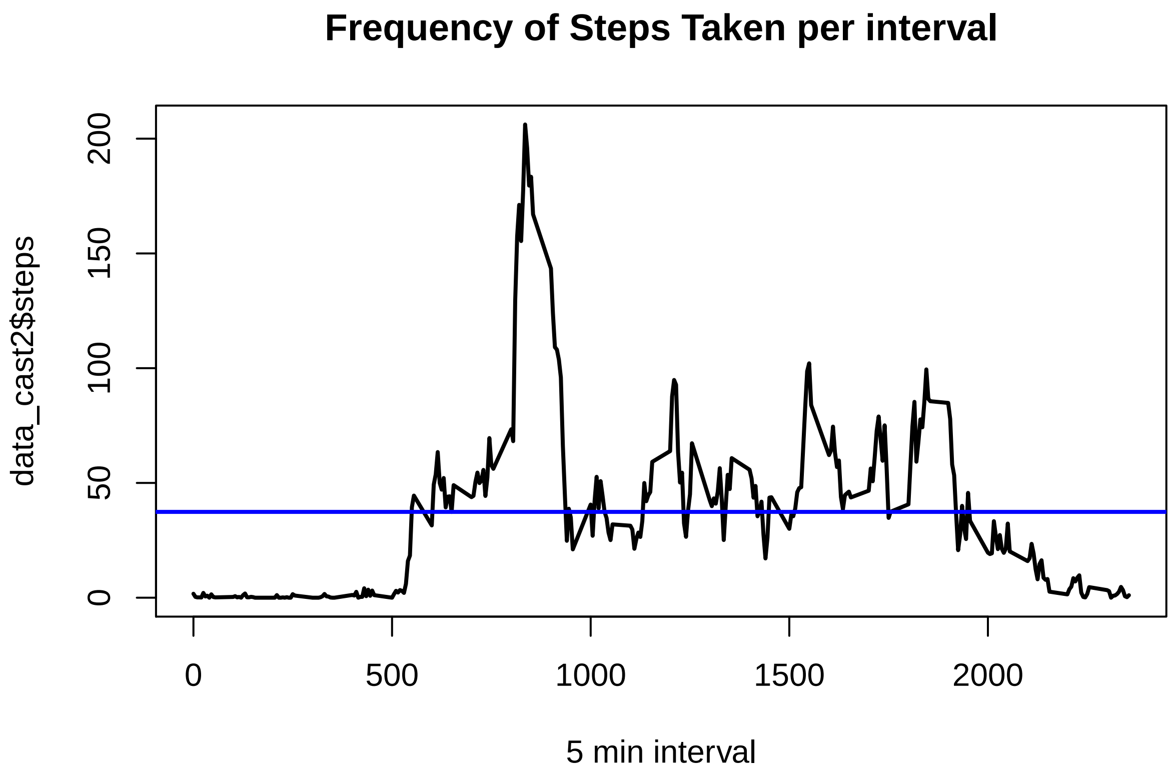
```
## [1] "Mean steps/day = 10766.1886792453"
```

```
paste("Median steps/day =", median(data_cast$steps, na.rm=TRUE))
```

```
## [1] "Median steps/day = 10765"
```

What is the average daily activity pattern?

```r
# Re-melt data frame to prearep for casting by interval. Now we include NA values
data_melt2 <- melt(data, id.vars="interval", measure.vars="steps", na.rm=TRUE)
# Cast data frame to see mean steps per interval
data_cast2 <- dcast(data_melt2, interval ~ variable, mean)
# Plot time series of average frequency of steps per interval
plot(data_cast2$interval, data_cast2$steps, type="l", main="Frequency of Steps Taken per interval",
xlab="5 min interval", lwd=2)
abline(h=mean(data_cast2$steps, na.rm=TRUE), col="blue", lwd=2)
```



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

```r
data_cast2[which.max(data_cast2$steps), ]
```

```
##     interval    steps
## 104      835 206.1698
```

Imput missing values

There are many 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. That is why we will input missing values by replacing them with the mean value.

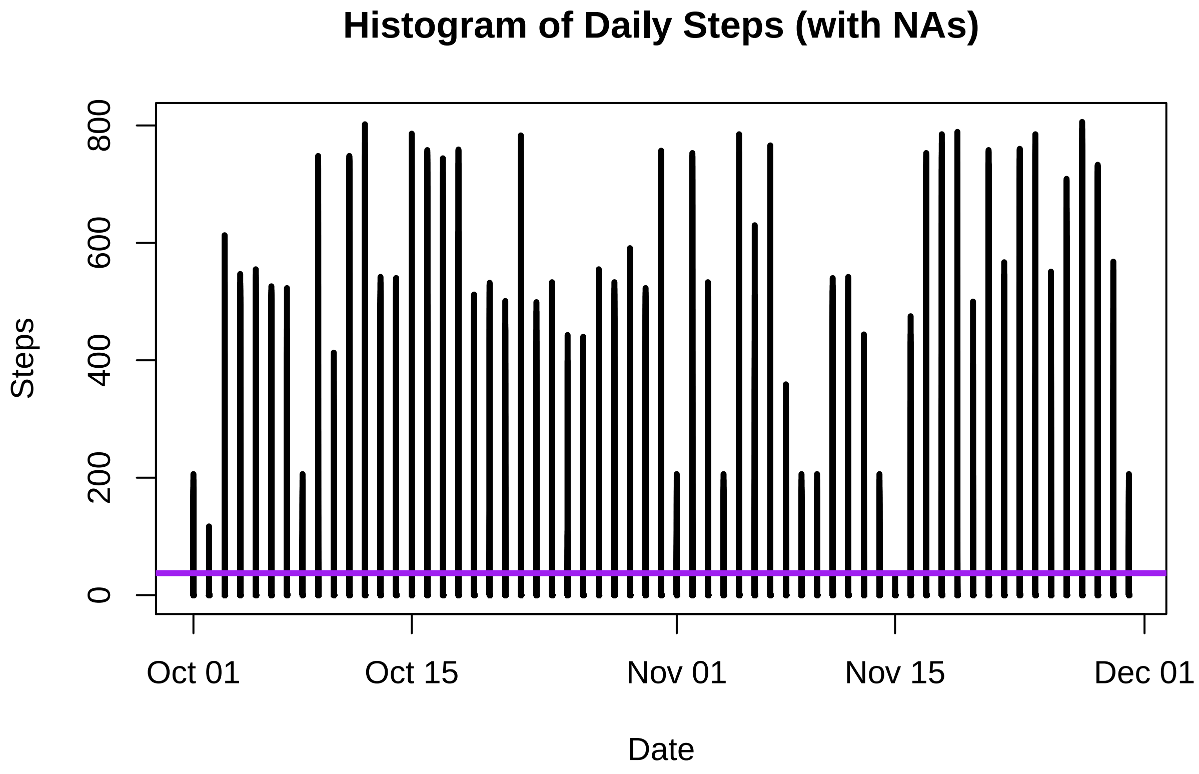
```
missing <- is.na(data$steps)
table(missing)
```

```
## missing
## FALSE  TRUE
## 15264  2304
```

```
# Replace each missing value with the mean value of its 5-minute interval
fill.value <- function(steps, interval) {
filled <- NA
if (!is.na(steps))
filled <- c(steps) else filled <- (data_cast2[data_cast2$interval == interval, "steps"])
return(filled)
}
filled.data <- data
filled.data$steps <- mapply(fill.value, filled.data$steps, filled.data$interval)
```

Histogram 2

```
#Now, we make a histogram using the filled data set with missing values.
plot(filled.data$date, filled.data$steps, type="h", main="Histogram of Daily Steps (with NAs)", xlab="Da
abline(h=mean(filled.data$steps), col="purple", lwd=3)
```



Calculate mean and median of daily steps

```
paste("Mean daily steps =", mean(filled.data$steps, na.rm=TRUE))
```

```
## [1] "Mean daily steps = 37.3825995807128"
```

```
paste("Median daily steps =", median(filled.data$steps, na.rm=TRUE))
```

```
## [1] "Median daily steps = 0"
```

Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

Mean and median values are higher after imputing missing data. The reason is that in the original data, there are some days with steps values NA for any interval. The total number of steps taken in such days are set to 0s by default but after replacing missing values with the mean, number of steps is not 0 any more, thus the estimates are higher after imputing NAs.

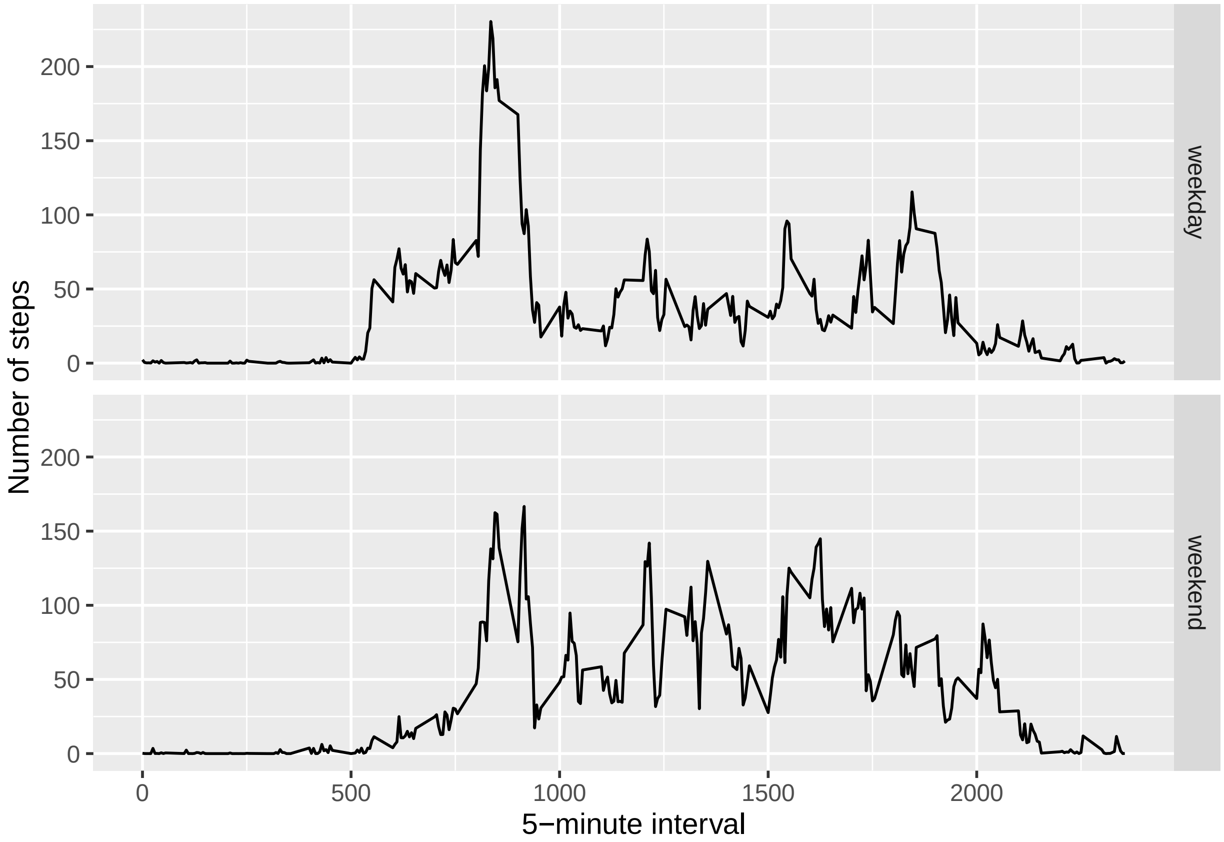
Are there differences in activity patterns between weekdays and weekends?

We first find the day of the week for each measurement. We use the dataset with the imputed NAs.

```
weekday.or.weekend <- function(date) {
day <- weekdays(date)
if (day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))
return("weekday") else if (day %in% c("Saturday", "Sunday"))
return("weekend") else stop("invalid date")
}
filled.data$date <- as.Date(filled.data$date)
filled.data$day <- sapply(filled.data$date, FUN = weekday.or.weekend)
```

Plot average number of steps taken on weekdays and weekends.

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


weekday
weekend
Number of steps
5−minute interval