Reproducible Research Assignment 1

James McCurdy

October 19, 2015

# Loading and preprocessing the data

Show any code that is needed to

1. **Load the data (i.e. read.csv()).**

# load the data  
activity <- read.csv("activity.csv")  
# check the data  
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 ...

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

Initially I did no processing. However, in the weekend/weekday step. I had to change it.

activity$date <- as.Date(activity$date, format = "%Y-%m-%d")

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

For this part of the assignment, you can ignore the missing values in the dataset.

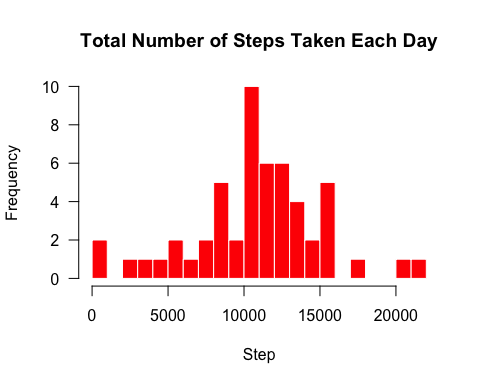
1. **Calculate the total number of steps taken per day.**

# Used aggregate fuction to add all the steps according date  
total\_step <- aggregate(steps ~ date, data = activity, sum, na.rm = TRUE)  
head(total\_step)

## date steps  
## 1 2012-10-02 126  
## 2 2012-10-03 11352  
## 3 2012-10-04 12116  
## 4 2012-10-05 13294  
## 5 2012-10-06 15420  
## 6 2012-10-07 11015

1. **If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day.**

par(mfrow = c(1, 1))  
# Used histogram, added in settings to make it look prettier.  
hist(total\_step$steps, breaks = 20,   
 main = "Total Number of Steps Taken Each Day",  
 col = "red", border = "white", xlab = "Step", axes = FALSE)  
axis(1)  
axis(2, las = 1)



1. **Calculate and report the mean and median of the total number of steps taken per day.**

mean(total\_step$steps)

## [1] 10766.19

median(total\_step$steps)

## [1] 10765

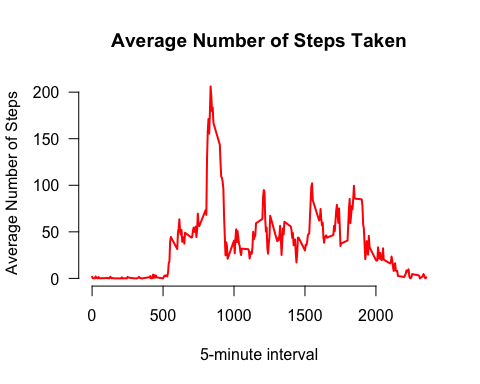
The mean is 10766 steps/day

The median is 10765 steps/day

# What is the average daily activity pattern?

1. **Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis).**

avg\_step <- aggregate(steps ~ interval, data = activity, mean, na.rm = TRUE)  
plot(avg\_step$interval, avg\_step$steps, type = "l", lwd = 2, col = "red",  
 main = "Average Number of Steps Taken", axes = FALSE,  
 xlab = "5-minute interval", ylab = "Average Number of Steps")  
axis(1)  
axis(2, las = 1)



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

avg\_step$interval[which.max(avg\_step$steps)]

## [1] 835

The 835 5m step interval contains the maximum number of steps.

# Inputing 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 NAs).**

#Sum of all of the missing values.  
sum(is.na(activity))

## [1] 2304

There are 2304 missing values in the dataset.

1. **Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.**

Used the mean for each 5 min interval. Replaced the missing elements with the mean of the containing interval.

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

imp <- activity # use a new name for activity, as to leave activity untouched as original   
for (i in avg\_step$interval) {  
 imp[imp$interval == i & is.na(imp$steps), ]$steps <-   
 avg\_step$steps[avg\_step$interval == i]  
}  
head(imp) # are there NAs?

## steps date interval  
## 1 1.7169811 2012-10-01 0  
## 2 0.3396226 2012-10-01 5  
## 3 0.1320755 2012-10-01 10  
## 4 0.1509434 2012-10-01 15  
## 5 0.0754717 2012-10-01 20  
## 6 2.0943396 2012-10-01 25

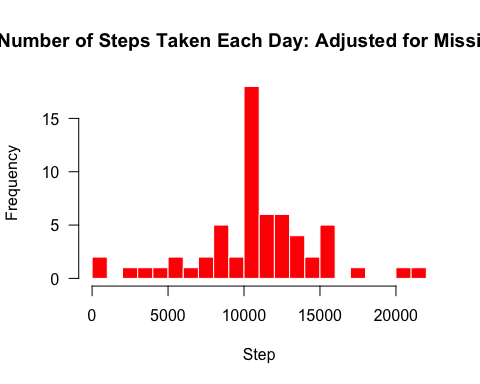
sum(is.na(imp)) # is the 0?

## [1] 0

It is 0, we're good.

1. **Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. 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?**

total\_step\_imp <- aggregate(steps ~ date, data = imp, sum, na.rm = TRUE)  
hist(total\_step\_imp$steps, breaks = 20,   
 main = "Total Number of Steps Taken Each Day: Adjusted for Missing Values",  
 col = "red", border = "white", xlab = "Step", axes = FALSE)  
axis(1)  
axis(2, las = 1)



mean(total\_step\_imp$steps)

## [1] 10766.19

median(total\_step\_imp$steps)

## [1] 10766.19

The mean is identical to the initial analysis. However the median has changed. In each 5m interval, there may be more missing values in various intervals. Since we are putting the missing values as the mean of each interval, it makes sense that the median becomes closer to the mean.

# Are there differences in activity patterns between weekdays and weekends?

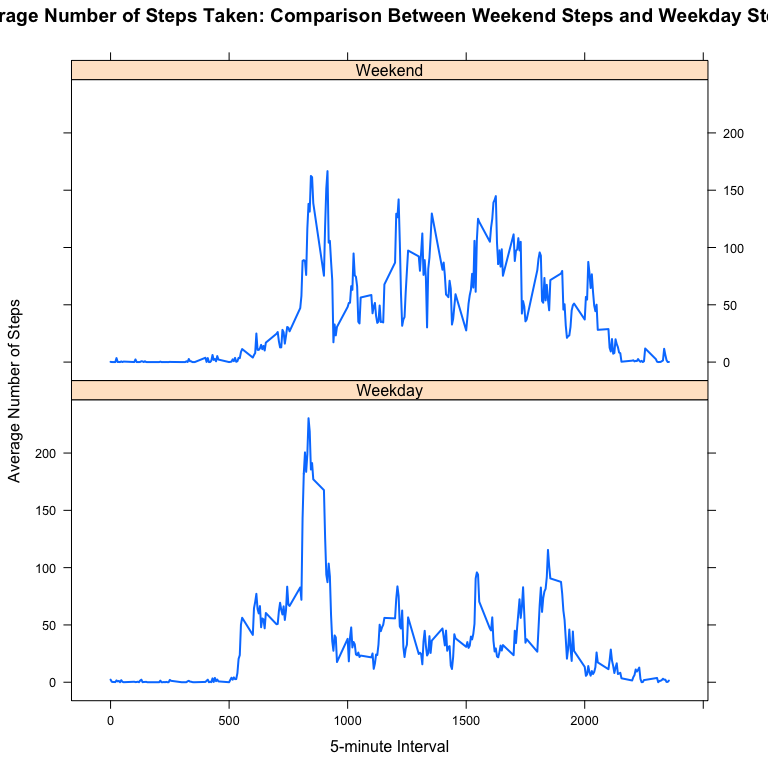
For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

1. **Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.**

imp$day <- weekdays(imp$date)  
imp$week <- ""  
imp[imp$day == "Saturday" | imp$day == "Sunday", ]$week <- "Weekend"  
imp[!(imp$day == "Saturday" | imp$day == "Sunday"), ]$week <- "Weekday"  
imp$week <- factor(imp$week)

1. **Make 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, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.**

avg\_step\_imp <- aggregate(steps ~ interval + week, data = imp, mean)  
library(lattice)  
xyplot(steps ~ interval | week, data = avg\_step\_imp, type = "l", lwd = 2,  
 layout = c(1, 2),   
 xlab = "5-minute Interval",   
 ylab = "Average Number of Steps",  
 main = "Average Number of Steps Taken: Comparison Between Weekend Steps and Weekday Steps.")



I like the plot. Makes for easy comparison.