Reproducible Research Assignment 1

Visu

Thursday, June 11, 2015

# Clear the workspace  
  
rm(list=ls())  
  
# Load the raw activity data  
activity\_raw <- read.csv("c:/coursera/data/activity.csv", stringsAsFactors=FALSE)  
  
# Transform the date attribute to an actual date format  
activity\_raw$date <- as.POSIXct(activity\_raw$date, format="%Y-%m-%d")  
  
# Compute the weekdays from the date attribute  
activity\_raw <- data.frame(date=activity\_raw$date,   
 weekday=tolower(weekdays(activity\_raw$date)),   
 steps=activity\_raw$steps,   
 interval=activity\_raw$interval)  
  
# Compute the day type (weekend or weekday)  
activity\_raw <- cbind(activity\_raw,   
 daytype=ifelse(activity\_raw$weekday == "saturday" |   
 activity\_raw$weekday == "sunday", "weekend",   
 "weekday"))  
  
# Create the final data.frame  
activity <- data.frame(date=activity\_raw$date,   
 weekday=activity\_raw$weekday,   
 daytype=activity\_raw$daytype,   
 interval=activity\_raw$interval,  
 steps=activity\_raw$steps)  
  
# Clear the workspace  
rm(activity\_raw)  
head(activity)

## date weekday daytype interval steps  
## 1 2012-10-01 monday weekday 0 NA  
## 2 2012-10-01 monday weekday 5 NA  
## 3 2012-10-01 monday weekday 10 NA  
## 4 2012-10-01 monday weekday 15 NA  
## 5 2012-10-01 monday weekday 20 NA  
## 6 2012-10-01 monday weekday 25 NA

What is the mean total number of steps taken per day? For this part of the assignment, you can ignore the missing values in the dataset.

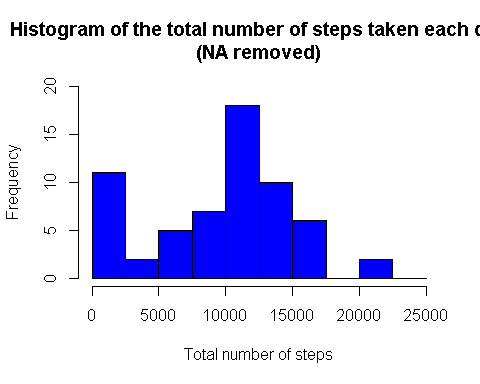
1. Make a histogram of the total number of steps taken each day

# Compute the total number of steps each day (NA values removed)  
sum\_data <- aggregate(activity$steps, by=list(activity$date), FUN=sum, na.rm=TRUE)  
  
# Rename the attributes  
names(sum\_data) <- c("date", "total")  
head(sum\_data)

## date total  
## 1 2012-10-01 0  
## 2 2012-10-02 126  
## 3 2012-10-03 11352  
## 4 2012-10-04 12116  
## 5 2012-10-05 13294  
## 6 2012-10-06 15420

The histogram is given by the following lines of code:

# Compute the histogram of the total number of steps each day  
hist(sum\_data$total,   
 breaks=seq(from=0, to=25000, by=2500),  
 col="blue",   
 xlab="Total number of steps",   
 ylim=c(0, 20),   
 main="Histogram of the total number of steps taken each day\n(NA removed)")



1. Calculate and report the mean and median total number of steps taken per day The mean and median are computed like

mean(sum\_data$total)

## [1] 9354.23

median(sum\_data$total)

## [1] 10395

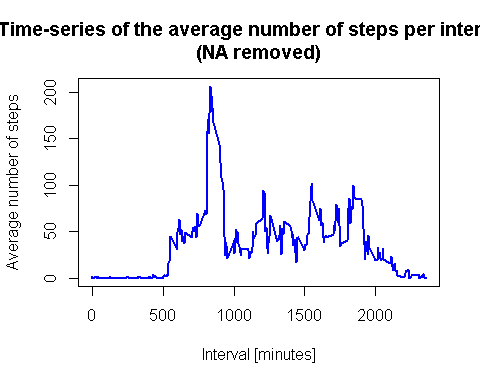
# What is the average daily activity pattern?

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)

# Clear the workspace  
rm(sum\_data)  
  
# Compute the means of steps accross all days for each interval  
mean\_data <- aggregate(activity$steps,   
 by=list(activity$interval),   
 FUN=mean,   
 na.rm=TRUE)  
  
# Rename the attributes  
names(mean\_data) <- c("interval", "mean")  
head(mean\_data)

## interval mean  
## 1 0 1.7169811  
## 2 5 0.3396226  
## 3 10 0.1320755  
## 4 15 0.1509434  
## 5 20 0.0754717  
## 6 25 2.0943396

# Compute the time series plot  
plot(mean\_data$interval,   
 mean\_data$mean,   
 type="l",   
 col="blue",   
 lwd=2,   
 xlab="Interval [minutes]",   
 ylab="Average number of steps",   
 main="Time-series of the average number of steps per intervals\n(NA removed)")



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

# We find the position of the maximum mean  
max\_pos <- which(mean\_data$mean == max(mean\_data$mean))  
  
# We lookup the value of interval at this position  
max\_interval <- mean\_data[max\_pos, 1]  
head(max\_interval)

## [1] 835

# Inputing the 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)

# Clear the workspace  
rm(max\_interval)  
  
# We use the trick that a TRUE boolean value is equivalent to 1 and a FALSE to 0.  
NA\_count <- sum(is.na(activity$steps))  
print(NA\_count)

## [1] 2304

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.

# Clear the workspace  
rm(NA\_count)  
  
# Find the NA positions  
na\_pos <- which(is.na(activity$steps))  
  
# Create a vector of means  
mean\_vec <- rep(mean(activity$steps, na.rm=TRUE), times=length(na\_pos))

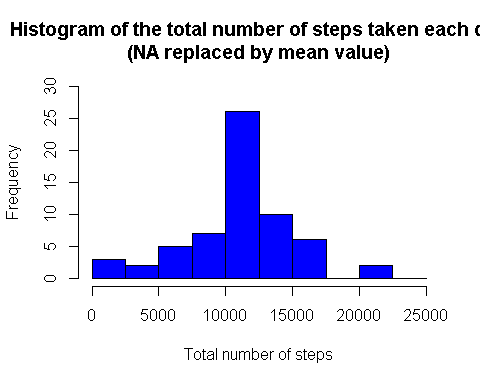
1. Create a new dataset that is equal to the original dataset but with the missing data filled in.

# Replace the NAs by the means  
activity[na\_pos, "steps"] <- mean\_vec  
  
# Clear the workspace  
rm(mean\_vec, na\_pos)  
head(activity)

## date weekday daytype interval steps  
## 1 2012-10-01 monday weekday 0 37.3826  
## 2 2012-10-01 monday weekday 5 37.3826  
## 3 2012-10-01 monday weekday 10 37.3826  
## 4 2012-10-01 monday weekday 15 37.3826  
## 5 2012-10-01 monday weekday 20 37.3826  
## 6 2012-10-01 monday weekday 25 37.3826

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?

# Compute the total number of steps each day (NA values removed)  
sum\_data <- aggregate(activity$steps, by=list(activity$date), FUN=sum)  
  
# Rename the attributes  
names(sum\_data) <- c("date", "total")  
  
# Compute the histogram of the total number of steps each day  
hist(sum\_data$total,   
 breaks=seq(from=0, to=25000, by=2500),  
 col="blue",   
 xlab="Total number of steps",   
 ylim=c(0, 30),   
 main="Histogram of the total number of steps taken each day\n(NA replaced by mean value)")



# The mean and median are computed like  
mean(sum\_data$total)

## [1] 10766.19

median(sum\_data$total)

## [1] 10766.19

These values differ greatly from the estimates from the first part of the assignment. The impact of imputing the missing values is to have more data, hence to obtain a bigger mean and median value.

## 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 - "weekdays" and "weekend" indicating whether a given date is a weekday or weekend day.

# The new factor variable "daytype" was already in the activity data frame  
head(activity)

## date weekday daytype interval steps  
## 1 2012-10-01 monday weekday 0 37.3826  
## 2 2012-10-01 monday weekday 5 37.3826  
## 3 2012-10-01 monday weekday 10 37.3826  
## 4 2012-10-01 monday weekday 15 37.3826  
## 5 2012-10-01 monday weekday 20 37.3826  
## 6 2012-10-01 monday weekday 25 37.3826

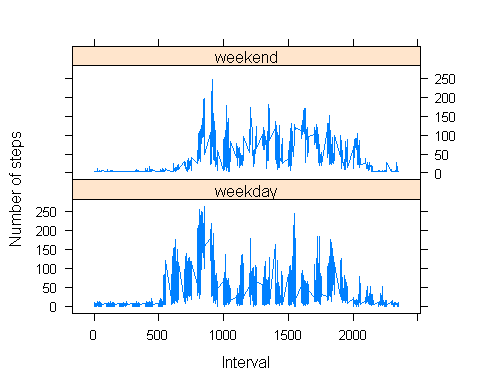
1. Make a panel plot containing a time series plot (i.e. type = "l") of the 5- minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

# Clear the workspace  
rm(sum\_data)  
  
# Load the lattice graphical library  
library(lattice)  
  
# Compute the average number of steps taken, averaged across all daytype variable  
mean\_data <- aggregate(activity$steps,   
 by=list(activity$daytype,   
 activity$weekday, activity$interval), mean)  
  
# Rename the attributes  
names(mean\_data) <- c("daytype", "weekday", "interval", "mean")  
head(mean\_data)

## daytype weekday interval mean  
## 1 weekday friday 0 8.307244  
## 2 weekday monday 0 9.418355  
## 3 weekend saturday 0 4.672825  
## 4 weekend sunday 0 4.672825  
## 5 weekday thursday 0 9.375844  
## 6 weekday tuesday 0 0.000000

The time series plot take the following form:

# Compute the time serie plot  
xyplot(mean ~ interval | daytype, mean\_data,   
 type="l",   
 lwd=1,   
 xlab="Interval",   
 ylab="Number of steps",   
 layout=c(1,2))



# Clear the workspace  
rm(mean\_data)