Class 5 Reproducible Research

## Assignment 1

This assignment 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.

The variables included in this dataset are:

steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)  
date: The date on which the measurement was taken in YYYY-MM-DD format  
interval: Identifier for the 5-minute interval in which measurement was taken

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

Submit:

1. Code for reading in the dataset and/or processing the data
2. Histogram of the total number of steps taken each day
3. Mean and median number of steps taken each day
4. Time series plot of the average number of steps taken
5. The 5-minute interval that, on average, contains the maximum number of steps
6. Code to describe and show a strategy for imputing missing data
7. Histogram of the total number of steps taken each day after missing values are imputed
8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends
9. All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

### 1. Code for reading in the dataset and/or processing the data

#### Loading and preprocessing the data

# Prepare the workspace  
rm(list=ls())  
  
# Set the working directory  
setwd("C:/Users/star/Desktop/COURSERA/Class\_5\_Reproducible\_Research\_Project\_1/Project\_1")  
  
# Load the raw data  
activity\_raw <- read.csv("activity.csv", stringsAsFactors=FALSE)  
  
# Check the the internal structure of an R object  
str(activity\_raw)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : chr "10/1/2012" "10/1/2012" "10/1/2012" "10/1/2012" ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

# Transform the date character to a date format  
activity\_raw$date <- as.Date(activity\_raw$date,format="%m/%d/%Y")  
str(activity\_raw)

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

head(activity\_raw)

## 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

# Compute the weekdays  
activity\_raw<-data.frame(date=activity\_raw$date, weekday=tolower(weekdays(activity\_raw$date)), steps=activity\_raw$steps,interval=activity\_raw$interval)  
  
head(activity\_raw)

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

# Compute the day type (weekend or weekday)  
activity\_raw<-cbind(activity\_raw, daytype=ifelse(activity\_raw$weekday=="saturday" | activity\_raw$weekday=="sunday", "weekend","weekday"))  
  
head(activity\_raw)

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

# Combine into 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)  
  
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

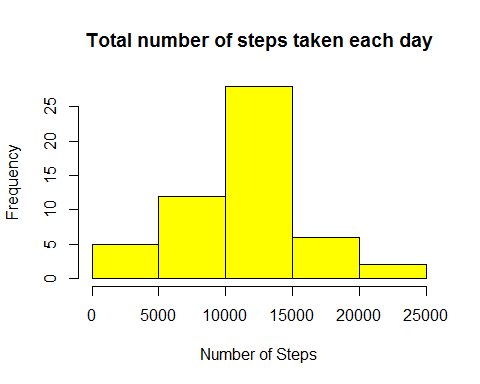
# Clear the workspace  
rm(activity\_raw)

### 2. Histogram of the total number of steps taken each day

steps\_per\_day<-aggregate(steps ~ date, activity, sum)  
head(steps\_per\_day)

## 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

hist(steps\_per\_day$steps, main = "Total number of steps taken each day", xlab = "Number of Steps", col = "yellow")



### 3. Mean and median number of steps taken each day

rmean <- mean(steps\_per\_day$steps)  
rmean

## [1] 10766.19

rmedian <- median(steps\_per\_day$steps)  
rmedian

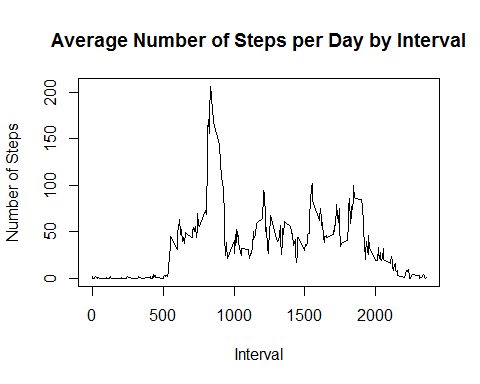
## [1] 10765

### 4. Time series plot of the average number of steps taken

steps\_per\_interval<-aggregate(steps ~ interval, activity, mean)  
head(steps\_per\_interval)

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

plot(steps\_per\_interval, type="l",main="Average Number of Steps per Day by Interval",xlab="Interval", ylab="Number of Steps")



### 5. The 5-minute interval that, on average, contains the maximum number of steps

#Maximum number of steps for a 5-minute interval  
maxSteps <- max(steps\_per\_interval$steps)  
maxSteps

## [1] 206.1698

#5-minute Interval with the maximum average number of steps  
steps\_per\_interval[steps\_per\_interval$steps==maxSteps,1]

## [1] 835

### 6. Code to describe and show a strategy for imputing missing data

#Count the total number of missing values in the dataset  
NA\_count <- sum(is.na(activity$steps))  
NA\_count

## [1] 2304

##As a strategy, you can count the mean of steps for each day and use instead of missing values  
  
# For that, find the NA positions  
na\_pos <- which(is.na(activity$steps))  
head(na\_pos)

## [1] 1 2 3 4 5 6

# Create a vector of means  
mean\_vec <- rep(mean(activity$steps, na.rm=TRUE), times=length(na\_pos))  
head(mean\_vec)

## [1] 37.3826 37.3826 37.3826 37.3826 37.3826 37.3826

##Replace each NA value by the mean of the steps attribute and create a new dataset with the missing data filled in.  
  
activity[na\_pos, "steps"] <- mean\_vec  
head(activity)

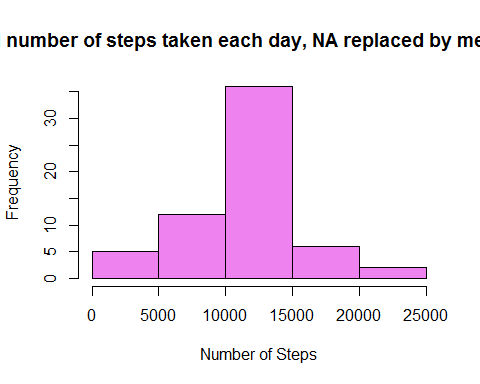
## date weekday daytype interval steps  
## 1 2012-10-01 monday weekday 0 37.3826  
## 2 2012-10-01 monday weekday 5 37.3826  
## 3 2012-10-01 monday weekday 10 37.3826  
## 4 2012-10-01 monday weekday 15 37.3826  
## 5 2012-10-01 monday weekday 20 37.3826  
## 6 2012-10-01 monday weekday 25 37.3826

### 7. Histogram of the total number of steps taken each day after missing values are imputed

sum\_data <- aggregate(steps ~ date, activity, sum)  
head(sum\_data)

## date steps  
## 1 2012-10-01 10766.19  
## 2 2012-10-02 126.00  
## 3 2012-10-03 11352.00  
## 4 2012-10-04 12116.00  
## 5 2012-10-05 13294.00  
## 6 2012-10-06 15420.00

hist(sum\_data$steps, main = "Total number of steps taken each day, NA replaced by mean value", xlab = "Number of Steps", col = "violet")



### 8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

#Average number of steps taken 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

library(lattice)  
  
xyplot(mean ~ interval | daytype, mean\_data,   
 type="l",  
 col="violet",  
 lwd=0.5,   
 xlab="Interval",   
 ylab="Number of steps",   
 layout=c(1,2))

