Reproducible Research: Peer Assessment 1

Uday Bhan Singh

Friday, February 13, 2015

```
  echo = TRUE  # Always make code visible
  options(scipen = 1)  # Turn off scientific notations for numbers

# R version 3.1.2 (2014-10-31)
  library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
  library(ggplot2)
  library(scales)

############ START OF QUESTION 1 ############

  ## Loading and preprocessing the data

  ## Q1. Show any code that is needed to :
  ##  1.  Load the data (i.e. read.csv())
  ##  2.  Process/transform the data (if necessary) into a format suitable for your analysis
  # Read unzipped csv file, define classes (according to assignment details) of column in it
  unzip("activity.zip")
  projFile <- read.csv("activity.csv", header = TRUE, stringsAsFactors=FALSE)

  # convert "interval" column into factors
  projFile$interval <- factor(projFile$interval)

  # change the date format in "date" column (i.e. YYYY-MM-DD)
  projFile$date <- as.Date(projFile$date, format = "%Y-%m-%d")

  # Structure of the dataset
  str(projFile)
```

```
## '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: Factor w/ 288 levels "0","5","10","15",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```
  # display first 6 rows from dataset
  head(projFile)
```

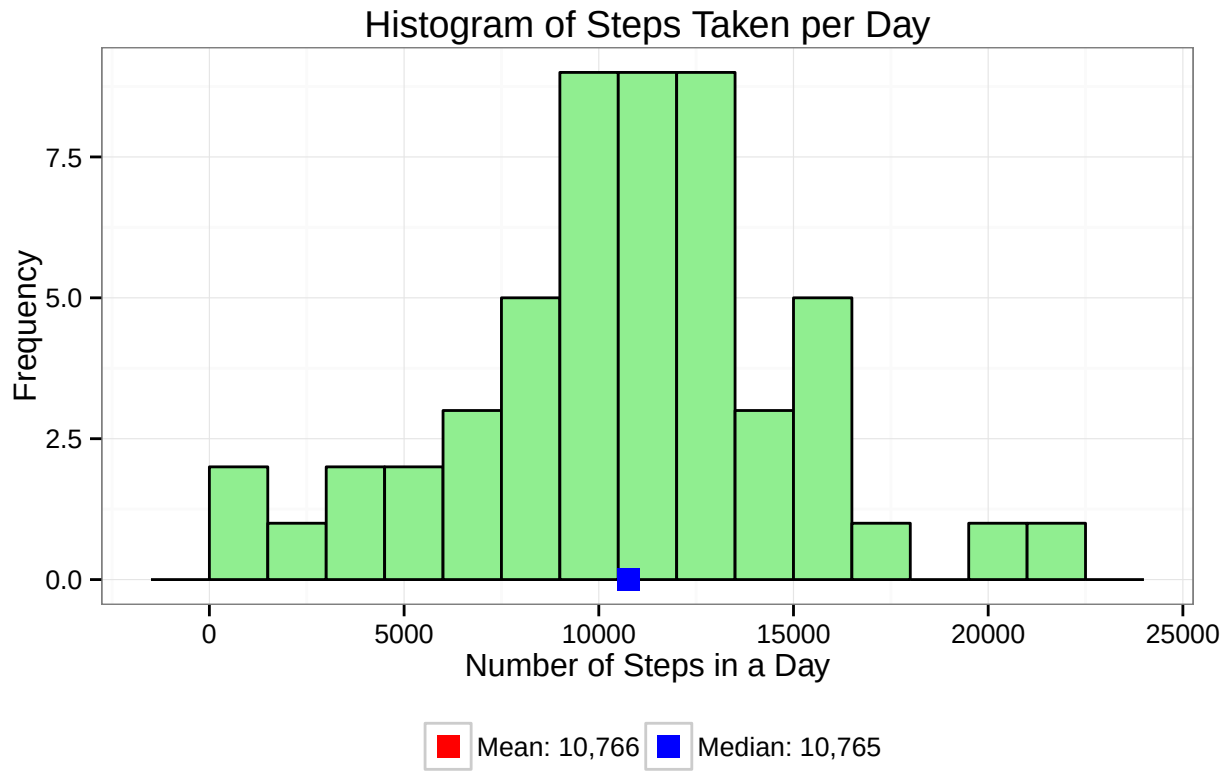
```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

```
  # Summary of the data
  summary(projFile)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   0      :   61
##  1st Qu.:  0.00   1st Qu.:2012-10-16   5      :   61
##  Median :  0.00   Median :2012-10-31   10     :   61
##  Mean   : 37.38   Mean   :2012-10-31   15     :   61
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   20     :   61
##  Max.   :806.00   Max.   :2012-11-30   25     :   61
##  NA's   :2304                          (Other):17202
```

```
############ START OF QUESTION 2 ############

  ## Q2. What is mean total number of steps taken per day?
  # Other question for which we have to provide solutions are:
  #   1. Make a histogram of the total number of steps taken each day
  #   2. Calculate and report the mean and median total number of steps taken per day

  # Part 1: Plotting Histogram
  plot_hist <- function(projFile) {
  plot_step <- aggregate(steps ~ date, projFile, sum)
  colnames(plot_step) <- c("date", "steps")
  plot_step
  }

  plot_format <- function(plot_step, proj_mean, proj_median) {
  clabs = c(paste("Mean:", formatC(proj_mean, big.mark = ",", format = "f", digits = 0)), paste("Median
  cols = c("red", "blue")

  ggplot(plot_step, aes(x = steps)) +
    geom_histogram(fill = "lightgreen", binwidth = 1500, color = "black") +
    geom_point(aes(x = proj_mean, y = 0, color = "blue"), size = 4, shape = 15) +
    geom_point(aes(x = proj_median, y = 0, color = "red"), size = 4, shape = 15) +
    scale_color_manual(name = element_blank(), labels = clabs, values = cols) +
    labs(title = "Histogram of Steps Taken per Day", x = "Number of Steps in a Day", y = "Frequency") +
    theme_bw() + theme(legend.position = "bottom")
}

  plot_step <- plot_hist(projFile)
  proj_mean = round(mean(plot_step$steps), 2)
```

```
  proj_median = round(median(plot_step$steps), 2)
  plot_format(plot_step, proj_mean, proj_median)
```




```
  # Part 2: Mean of the dataset
  paste("Mean total number of steps taken per day: ", round(proj_mean, 0), sep = " ")
```

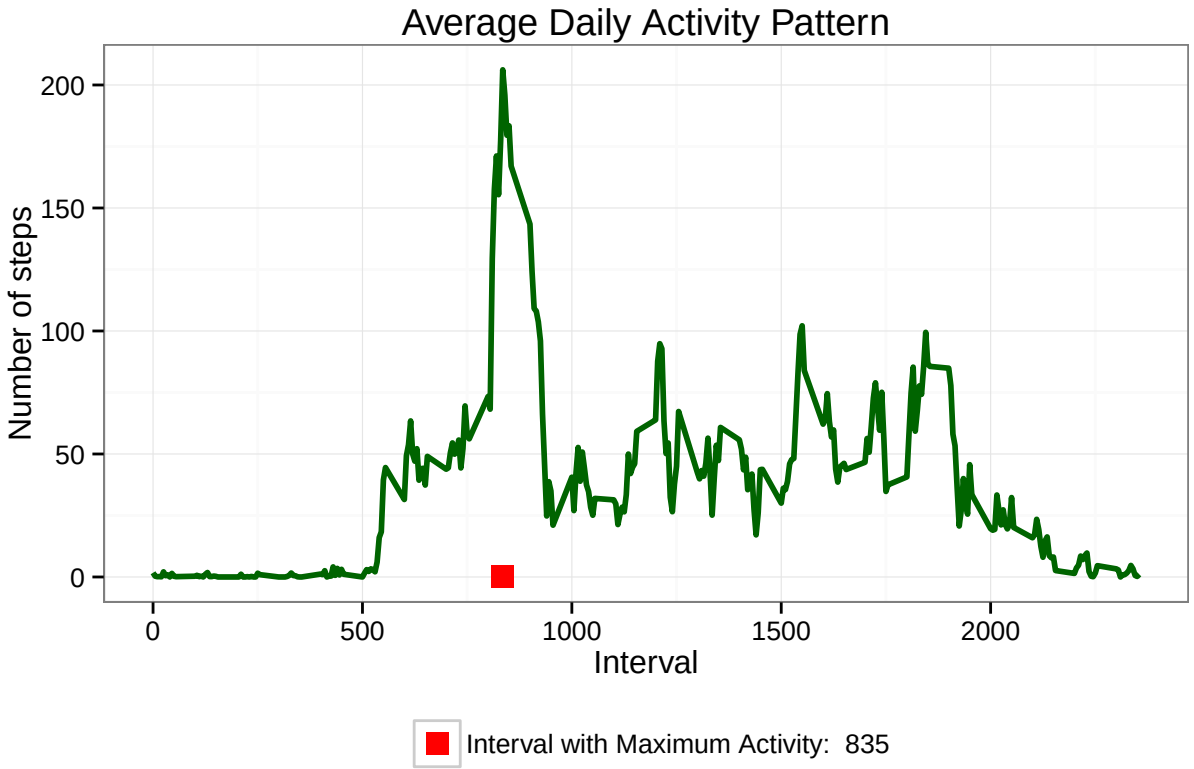
```
## [1] "Mean total number of steps taken per day:  10766"
```

```
  # Median of the dataset
  paste("Median total number of steps taken per day: ", round(proj_median, 0), sep = " ")
```

```
## [1] "Median total number of steps taken per day:  10765"
```

```
############ START OF QUESTION 3 ############

  ## Q3. 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 numb
  # 2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum num

  cspi <- function(projFile) {
    spi <- aggregate(projFile$steps, by = list(interval = projFile$interval),
                          FUN = mean, na.rm = TRUE)
```

```
  # convert to integers for plotting
  spi$interval <- as.integer(levels(spi$interval)[spi$interval])
  colnames(spi) <- c("interval", "steps")
  spi
}
pap <- function(spi, msi) {
clabs = c(paste("Interval with Maximum Activity: ", msi))
  cols = c("red")
  ggplot(spi, aes(x = interval, y = steps)) +
      geom_line(color = "darkgreen", size = 1) +
      geom_point(aes(x = msi, y = 0, color = "red"), size = 4, shape = 15) +
      scale_color_manual(name = element_blank(), labels = clabs, values = cols) +
      labs(title = "Average Daily Activity Pattern", x = "Interval", y = "Number of steps") +
      theme_bw() + theme(legend.position = "bottom")
}

spi <- cspi(projFile)
msi <- spi[which.max(spi$steps),]$interval
pap(spi, msi)
```



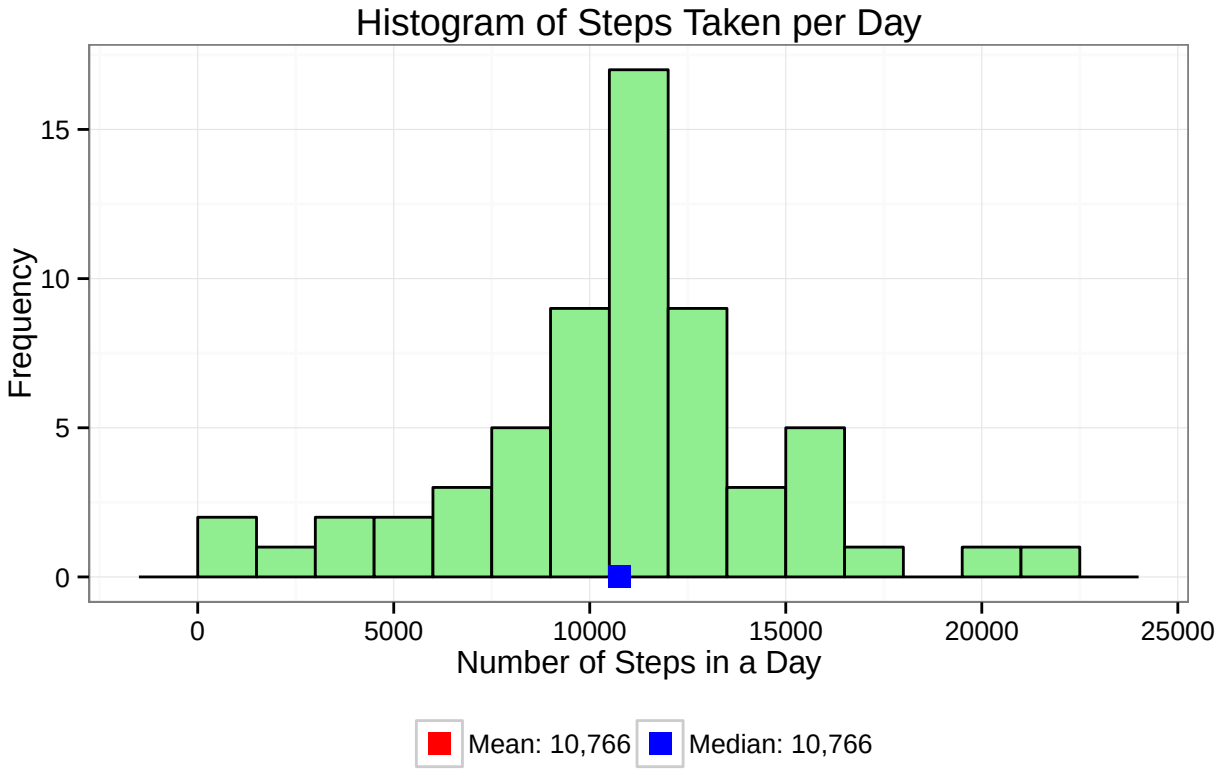

```
paste("Interval with Maximum Activity : ", msi, sep = " ")
```

```
## [1] "Interval with Maximum Activity :  835"
```

```
############ START OF QUESTION 4 #############

  # Imputing missing values

  # Note that there are a number of days/intervals where there are missing values (coded as NA). The pr

  # 1. Calculate and report the total number of missing values in the dataset (i.e. the total number of

  # 2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not

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

  # 4. Make a histogram of the total number of steps taken each day and Calculate and report the mean a


  imeans <- function(projFile, defs) {
    nai <- which(is.na(projFile$steps))
    defs <- spi
    nar <- unlist(lapply(nai, FUN = function(idx){
        interval = projFile[idx, ]$interval
        defs[defs$interval == interval, ]$steps
        }))
    imps <- projFile$steps
    imps[nai] <- nar
    imps
  }

  comt <- data.frame(
    steps = imeans(projFile, spi),
    date = projFile$date,
    interval = projFile$interval)


  summary(comt)
```

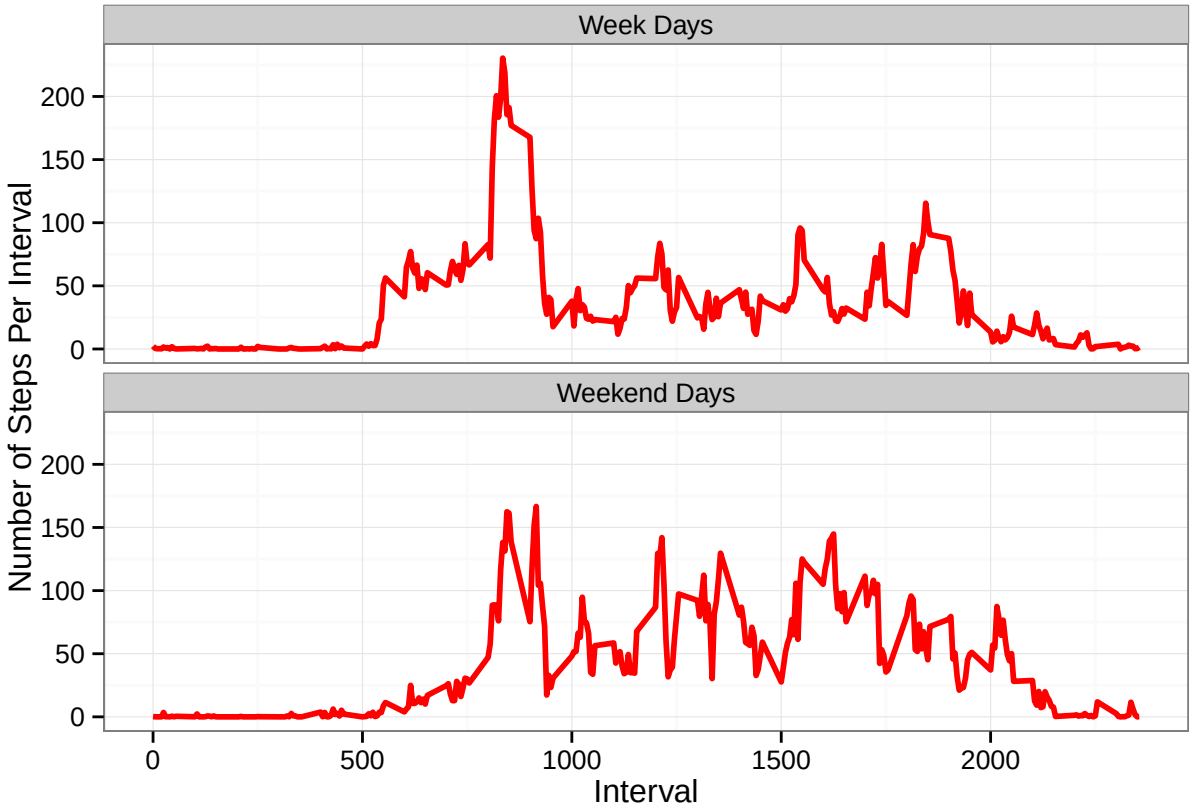
```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   0      :   61
##  1st Qu.:  0.00   1st Qu.:2012-10-16   5      :   61
##  Median :  0.00   Median :2012-10-31   10     :   61
##  Mean   : 37.38   Mean   :2012-10-31   15     :   61
##  3rd Qu.: 27.00   3rd Qu.:2012-11-15   20     :   61
##  Max.   :806.00   Max.   :2012-11-30   25     :   61
##                                        (Other):17202
```

```
  after_imp <- plot_hist(comt)
  after_imp_mean <- round(mean(after_imp$steps), 2)
  after_imp_median <- round(median(after_imp$steps), 2)
  plot_format(after_imp, after_imp_mean, after_imp_median)
```




```
############ START OF QUESTION 5 ############
  # Are there differences in activity patterns between weekdays and weekends?

  # First replace the missing values in the table.
  # Then we augment the table with a column that indicates the day of the week
  # Followed by subsetting the table into two parts -
  # weekends (Saturday and Sunday); and
  # weekdays (Monday through Friday)
  # We then tabulate the average steps per interval for each dataset.
  # And plot the two datasets side by side for comparison.

  cdwd <- function(projFile) {
    projFile$weekday <- as.factor(weekdays(projFile$date))

    # Subset of weekend days
    wed <- subset(projFile, weekday %in% c("Saturday", "Sunday"))

    # Subset of week days
    wkd <- subset(projFile, !weekday %in% c("Saturday", "Sunday"))

    wspi <- cspi(wed)
    wdspi <- cspi(wkd)

    wspi$dow <- rep("Weekend Days", nrow(wspi))
    wdspi$dow <- rep("Week Days", nrow(wdspi))
```

```
  dwd <- rbind(wspi, wdspi)
  dwd$dow <- as.factor(dwd$dow)
  dwd
}

pdwc <- function(dowd) {
  ggplot(dowd,
      aes(x = interval, y = steps)) +
      geom_line(color = "red", size = 1) +
      facet_wrap(~ dow, nrow = 2, ncol = 1) +
      labs(x = "Interval", y = "Number of Steps Per Interval") +
      theme_bw()
}

dofwd <- cdwd(comt)
pdwc(dofwd)
```




```
# As compare to weekends, the activity on the week days are widely spread.

# The obvious reason for that is on weekdays there are much more routine movements (due to work etc.)
```