Reproducible Project 1

Prasad P

29 October 2018

## R Markdown

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

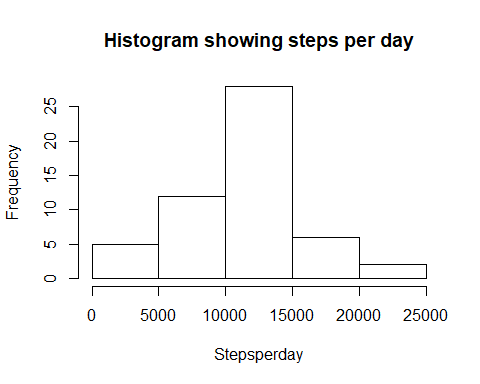
actv\_file <- read.csv("C:/Users/prasad.pande/Documents/activity.csv", header = TRUE)

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

Stepsperday <- tapply(actv\_file$steps, actv\_file$date, sum)  
summary(Stepsperday)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 41 8841 10765 10766 13294 21194 8

hist(Stepsperday, main = "Histogram showing steps per day")



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

Meanperday <- mean(Stepsperday, na.rm=TRUE)  
Meanperday

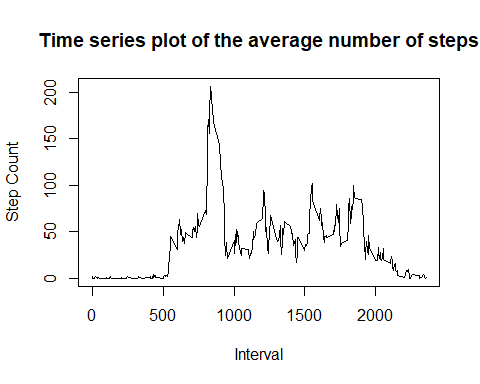
## [1] 10766.19

Medianperday <- median(Stepsperday, na.rm = TRUE)  
Medianperday

## [1] 10765

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

intervalsteps <- tapply(actv\_file$steps, actv\_file$interval, mean, na.rm = TRUE)  
#colnames(intervalsteps) <- c("Steps", "Interval")  
plot(as.numeric(names(intervalsteps)), intervalsteps, xlab="Interval", ylab="Step Count",type = "l", main = "Time series plot of the average number of steps")

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

maxsteps <- names(sort(intervalsteps, decreasing = TRUE)[1])  
maxsteps

## [1] "835"

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

library(tidyr)  
#tidyr package is used for using replace\_NA. I will be replacing NA by mean of steps  
actv\_file$steps <- actv\_file$steps %>% replace\_na(as.integer(mean(actv\_file$steps, na.rm = TRUE)))  
summary(actv\_file)

## 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 : 37.33 2012-10-04: 288 Mean :1177.5   
## 3rd Qu.: 37.00 2012-10-05: 288 3rd Qu.:1766.2   
## Max. :806.00 2012-10-06: 288 Max. :2355.0   
## (Other) :15840

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

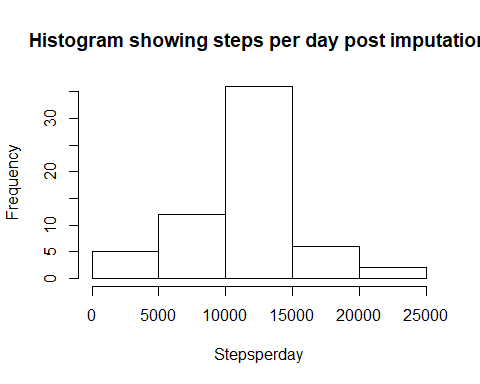
class(actv\_file)

## [1] "data.frame"

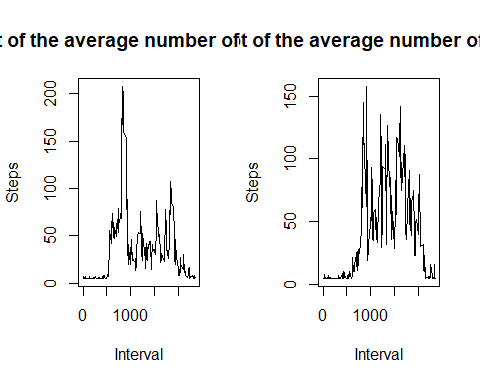
Stepsperday <- tapply(actv\_file$steps, actv\_file$date, sum)  
summary(Stepsperday)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41 9819 10656 10752 12811 21194

hist(Stepsperday, main = "Histogram showing steps per day post imputation")

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

actv\_file.date1 <- ifelse(weekdays(as.Date(actv\_file$date))== "Saturday"|weekdays(as.Date(actv\_file$date))== "Sunday","Weekend", "Weekday")  
  
actv\_file.weekend <- tapply(actv\_file[actv\_file.date1 == "Weekend",]$steps,  
 actv\_file[actv\_file.date1 == "Weekend",]$interval,  
 mean, na.rm=TRUE)  
actv\_file.weekday <- tapply(actv\_file[actv\_file.date1 == "Weekday",]$steps,  
 actv\_file[actv\_file.date1 == "Weekday",]$interval,  
 mean, na.rm=TRUE)  
# setting the 1 rows and 2 column plot  
par(mfrow=c(1,2))  
  
#Time series plot of the average number of steps on Weekdays  
plot(as.numeric(names(actv\_file.weekday)), actv\_file.weekday, type = "l",  
 xlab = "Interval", ylab = "Steps", main = "Time series plot of the average number of steps on Weekdays" )  
  
#Time series plot of the average number of steps on Weekend  
plot(as.numeric(names(actv\_file.weekend)), actv\_file.weekend, type = "l",  
 xlab = "Interval", ylab = "Steps", main = "Time series plot of the average number of steps on Weekend" )



# 9. All of the R code needed to reproduce the results (numbers, plots, etc.) in the report