Reproducible Research: Course Project 1

Analysis of Steps data

1. Loading data

The very first step of analysis is to have the right data on the device. I have downloaded the 'zip' file provided to us by Course instructors and placed it in a directory called **data**. So let's set the data directory accordingly.

```
wd <- getwd()
data_dir <- paste0(wd, '/data')
print(getwd())
```

```
## [1] "C:/Users/Shashwat Kadam/Documents/Coursera_DataScience/5_Reproducible_Research/Week2"
```

Now let's unzip the zip file and load the contents in the R object, using `read.csv()`

```
unzip(paste0(data_dir, '/repdata_data_activity.zip'))
data <- read.csv(paste0(data_dir, '/activity.csv'))
summary(data)
```

```
##      steps            date              interval     
##  Min.   :  0.00   Length:17568       Min.   :   0.0  
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8  
##  Median :  0.00   Mode  :character   Median :1177.5  
##  Mean   : 37.38                      Mean   :1177.5  
##  3rd Qu.: 12.00                      3rd Qu.:1766.2  
##  Max.   :806.00                      Max.   :2355.0  
##  NA's   :2304
```

Names of the columns are:

```
names(data)
```

```
## [1] "steps"    "date"     "interval"
```

2. Analysis

Load required libraries

```
library(dplyr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

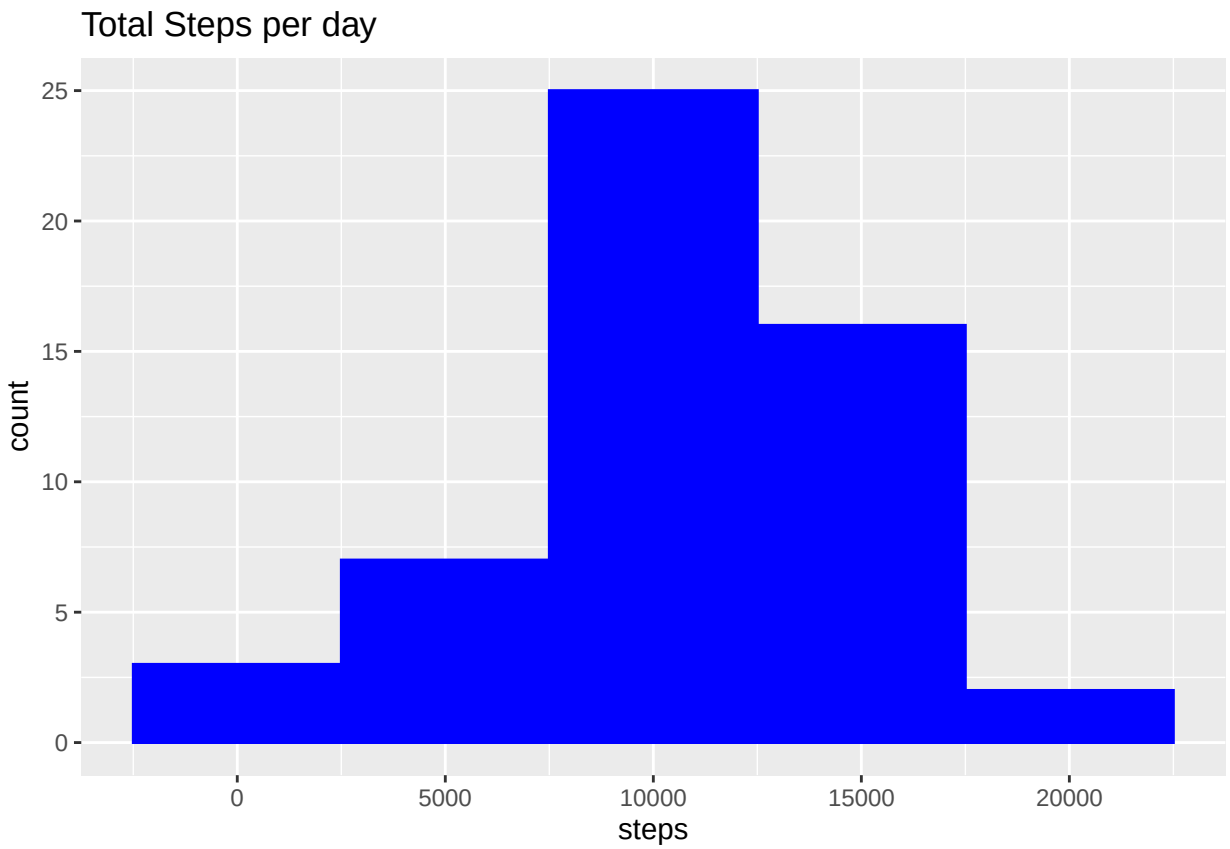
2.1. What are the average no. of steps per day?

- Calculate total steps per day

```
stepsPerDay <- aggregate(steps ~ date, data, sum, na.rm=TRUE)
# stepsPerDay <- data %>% group_by(date) %>% summarize(steps=sum(steps, na.rm=TRUE))
# Above line also produces same results, but somehow introduces 'NaN'
```

- Visualize using Histogram

```
p <- ggplot(stepsPerDay, aes(x=steps)) + geom_histogram(binwidth=5000, color='blue', fill='blue')+ ggti
print(p)
```



* Mean and median no. of steps

```
mean_steps <- mean(stepsPerDay$steps)
median_steps <- median(stepsPerDay$steps)
mean_steps
```

```
## [1] 10766.19
```

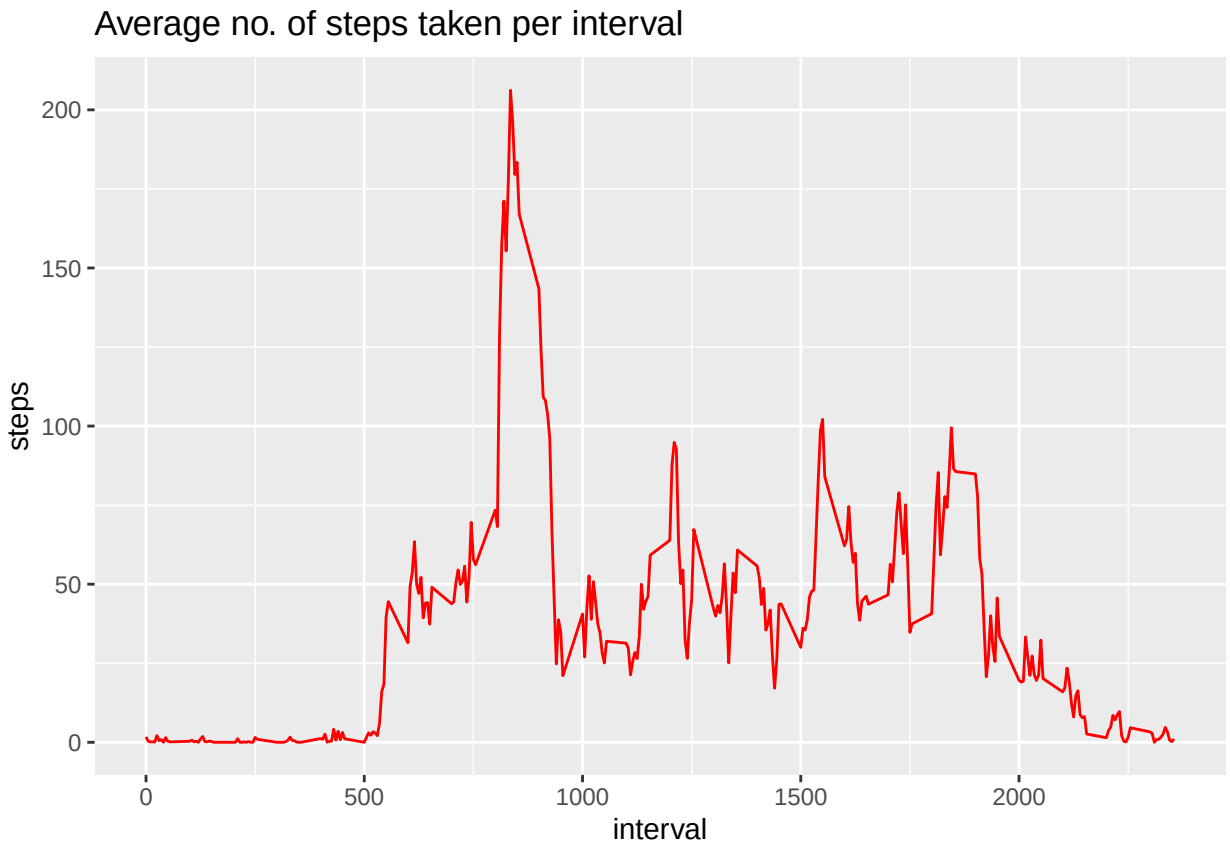
```
median_steps
```

```
## [1] 10765
```

2.2. Time series plot of Average no. of steps

```
stepsPerInterval <- aggregate(steps ~ interval, data, mean, na.rm=TRUE)
```

```
time_series <- ggplot(stepsPerInterval, aes(x=interval, y=steps)) + geom_line(color='red') + ggtitle("A
time_series
```



2.3. The five minute interval which contains maximum no. of steps

```
maxStepInterval <- stepsPerInterval[which.max(stepsPerInterval$steps),]$interval
maxStepInterval
```

```
## [1] 835
```

2.4. Imputing Missing data

Are there any missing values? Let's find out

```
sapply(names(data), function(x) sum(is.na(data[[x]])))
```

```
##    steps     date interval
##     2304        0        0
```

As you can see above, there are missing values in *steps* column, whereas no missing values are present in other two columns.

Let's fill the missing values using *mean* no. of steps in the interval

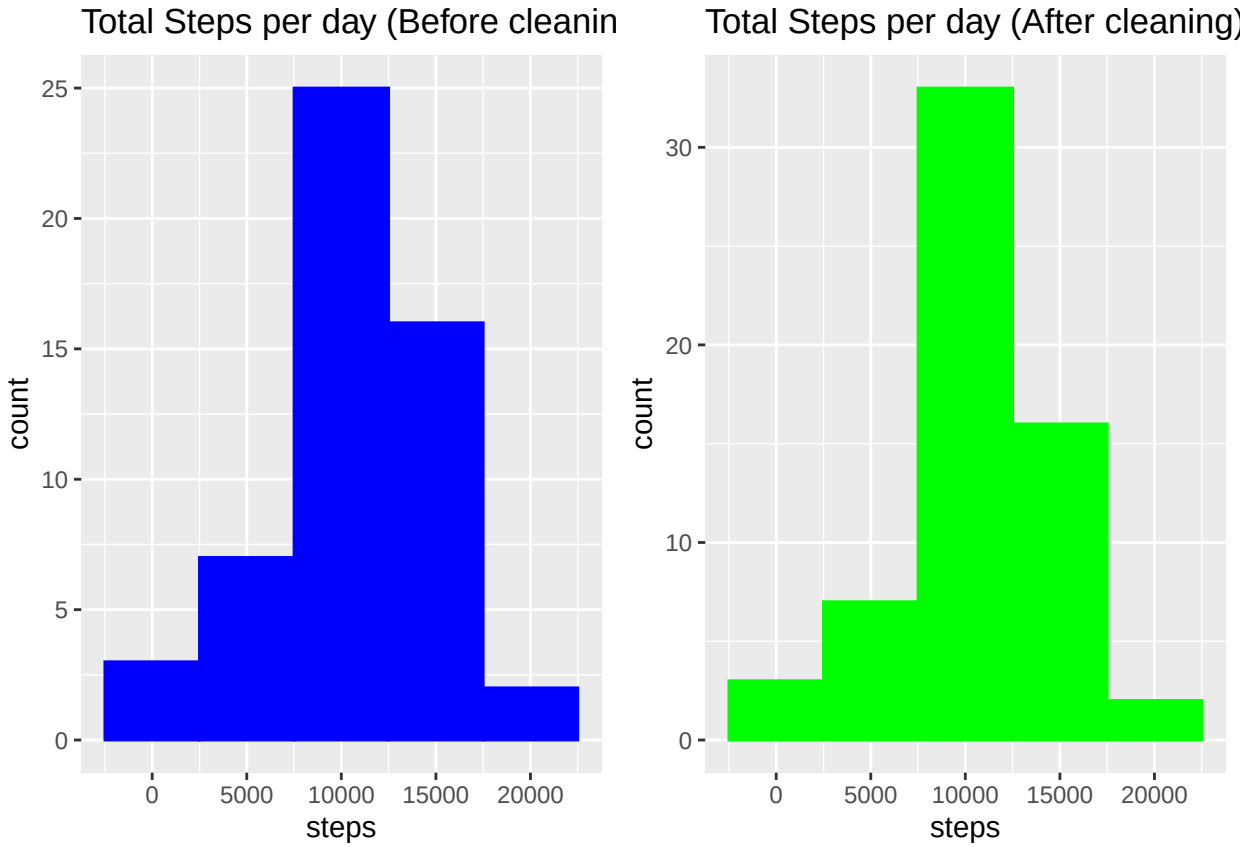
```
get_mean_of_interval <- function(interval) {
    stepsPerInterval[stepsPerInterval$interval==interval, 'steps']
}
clean_data <- data
for(i in 1:nrow(clean_data)){
    if(is.na(clean_data[i, 'steps'])){
        clean_data[i, 'steps'] <- get_mean_of_interval(clean_data[i, 'interval'])
```

```
    }
}
stepsPerDay2 <- aggregate(steps ~ date, clean_data, sum)
```

Once the data is cleaned, let's check whether the resulting histogram is differing from the previous one (Section 2.1).

```
require(gridExtra)
p1 <- ggplot(stepsPerDay, aes(x=steps)) + geom_histogram(binwidth=5000, color='blue', fill='blue')+ ggt

p2 <- ggplot(stepsPerDay2, aes(x=steps)) + geom_histogram(binwidth=5000, color='green', fill='green')+
grid.arrange(p1, p2, ncol=2)
```




```
mean_steps_cleaned <- mean(stepsPerDay2$steps)
median_steps_cleaned <- median(stepsPerDay2$steps)
# Difference
percent_change_mean <- ((mean_steps_cleaned - mean_steps)/mean_steps)*100
percent_change_median <- ((median_steps_cleaned - median_steps)/median_steps)*100
```

```
percent_change_mean
```

```
## [1] 0
```

```
percent_change_median
```

```
## [1] 0.01104207
```

2.5. Differences in activity on *weekdays* and *weekends*

We need an extra nominal variable (factor) which tells whether a particular day is **weekend** or **weekday**

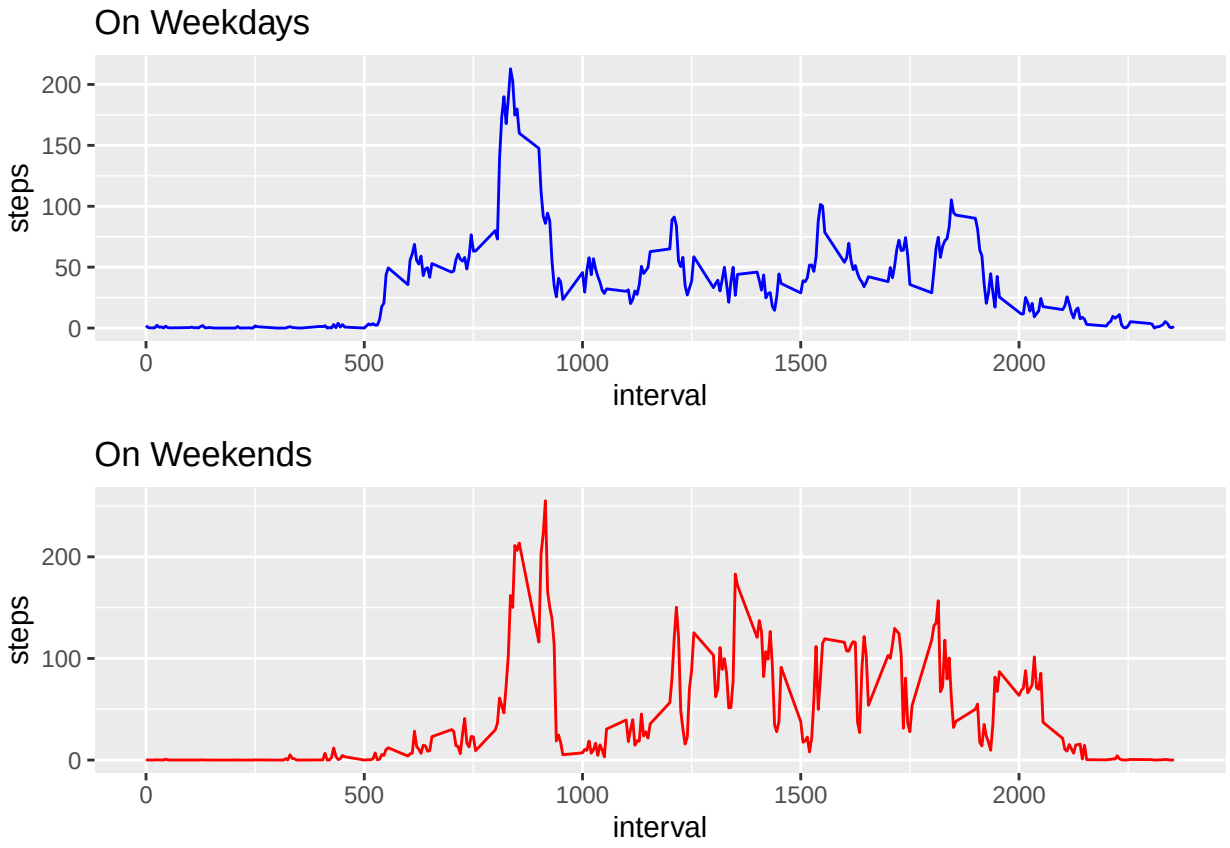
```
clean_data$date <- as.Date(strptime(clean_data$date, format="%Y-%m-%d"))
clean_data$day_type <- as.POSIXlt(clean_data$date)$wday
for (i in 1:nrow(clean_data)) {
    if (clean_data[i, 'day_type'] %in% c(6, 7)) {
        clean_data[i, 'day_type'] <- "weekend"
    }
    else{
        clean_data[i, 'day_type'] <- "weekday"
    }
}
stepsByDayType <- aggregate(steps ~ interval + day_type, clean_data, mean)
```

Now let's plot the difference using ggplot2

```
names(stepsByDayType)
```

```
## [1] "interval" "day_type" "steps"
```

```
weekday_plot <- ggplot(stepsByDayType[stepsByDayType$day_type=='weekday', ], aes(x=interval, y=steps)) +
weekend_plot <- ggplot(stepsByDayType[stepsByDayType$day_type=='weekend', ], aes(x=interval, y=steps)) +
grid.arrange(weekday_plot, weekend_plot, nrow=2)
```



There is more activity on weekends after interval **1000** evidencing that the persons might be engaging in outdoor activities like walking, trekking, etc.