Activity Monitoring

title: “Activity Monitoring” output: html_document —

Loading and preprocessing the data

we will start loading the data and transforming it if necessary,we will also observe the data variables

```
activity <- read.csv("activity.csv")
head(activity)
```

```
##   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
```

```
class(activity$date)
```

```
## [1] "character"
```

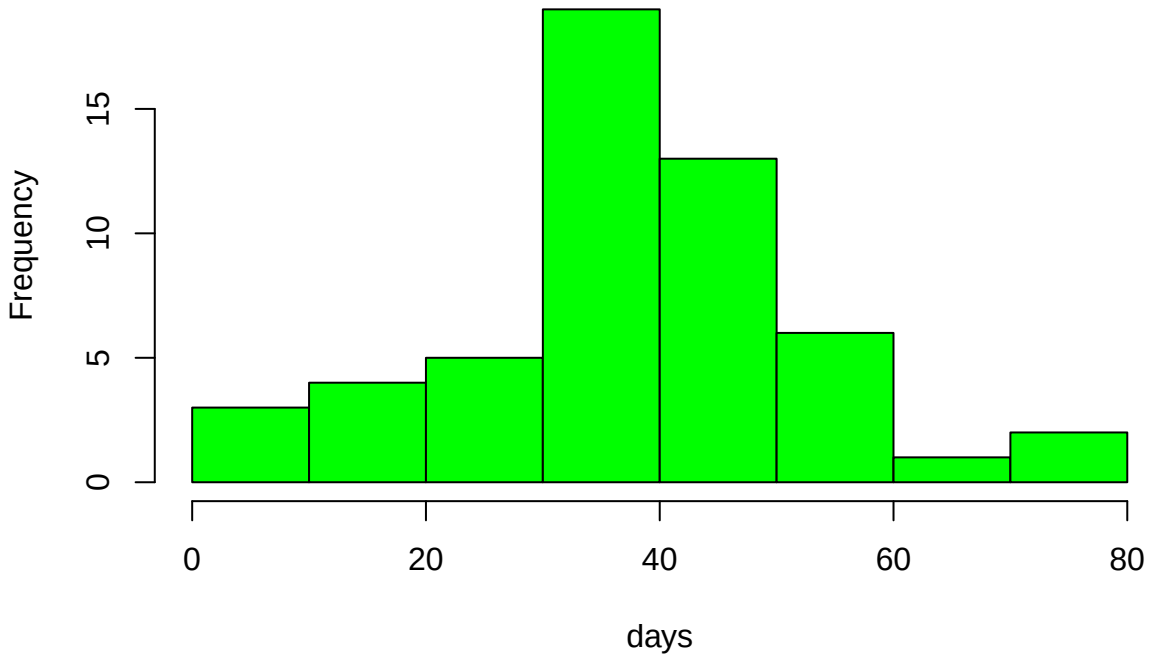
```
class(activity$interval)
```

```
## [1] "integer"
```

What is mean total number of steps taken per day?

we will first use a histogram to see the average frequency of steps per day and then we will create a data set of the mean and median steps per day

```
hist(tapply(activity$steps, activity$date, mean),
     main = "mean total number of steps per day",
     xlab = "days", col = "green")
```




```
library(plyr)
dfmm <- ddply(activity,.(date),summarize,
                mean = mean(steps))
dfmm$median <- ddply(activity,.(date),summarize,
                     median = median(steps))
head(dfmm,10)
```

```
##          date     mean median.date median.median
## 1  2012-10-01       NA  2012-10-01            NA
## 2  2012-10-02  0.43750  2012-10-02             0
## 3  2012-10-03 39.41667  2012-10-03             0
## 4  2012-10-04 42.06944  2012-10-04             0
## 5  2012-10-05 46.15972  2012-10-05             0
## 6  2012-10-06 53.54167  2012-10-06             0
## 7  2012-10-07 38.24653  2012-10-07             0
## 8  2012-10-08       NA  2012-10-08            NA
## 9  2012-10-09 44.48264  2012-10-09             0
## 10 2012-10-10 34.37500  2012-10-10             0
```

```
head(dfmm$median,10)
```

```
##          date median
## 1  2012-10-01     NA
## 2  2012-10-02      0
## 3  2012-10-03      0
## 4  2012-10-04      0
## 5  2012-10-05      0
## 6  2012-10-06      0
## 7  2012-10-07      0
## 8  2012-10-08     NA
## 9  2012-10-09      0
```

```
## 10 2012-10-10      0
```

What is the average daily activity pattern?

Next we are going to load the dplyr and lattice packages to create a summary table of the average of steps taken per day and interval and then graph the variables of interest

```
library(lattice)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
table1 <- ddply(activity,.(date),summarize,
                steps = mean(steps))

table2 <- ddply(activity,.(interval), summarise,
                 steps = max(summary(steps)))
```

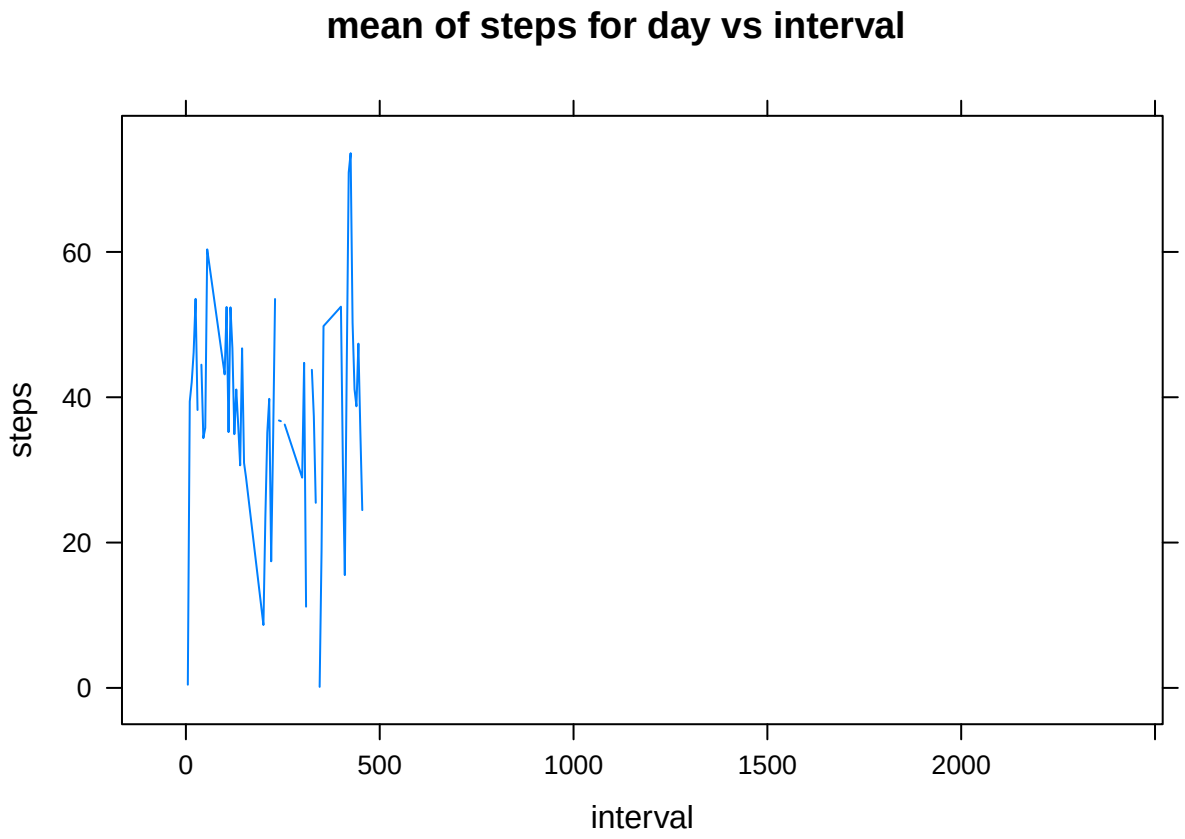
```
head(table1)
```

```
##         date    steps
## 1 2012-10-01       NA
## 2 2012-10-02  0.43750
## 3 2012-10-03 39.41667
## 4 2012-10-04 42.06944
## 5 2012-10-05 46.15972
## 6 2012-10-06 53.54167
```

```
head(table2)
```

```
##   interval steps
## 1        0    47
## 2        5    18
## 3       10     8
## 4       15     8
## 5       20     8
## 6       25    52
```

```
xyplot(table1$steps ~ table2$interval, type = "l",
       main = "mean of steps for day vs interval",
       xlab = "interval", ylab = "steps")
```



we will create a third table to find the interval with the greatest number of steps

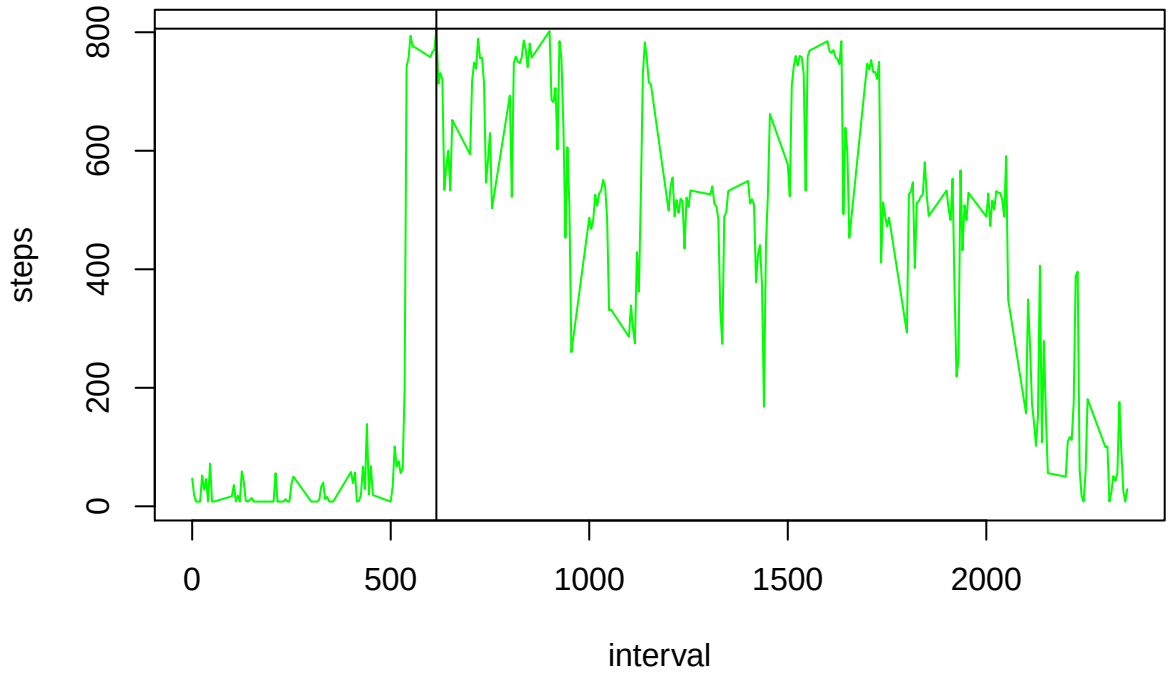
```
table3 <- arrange(table2, steps)
```

```
tail(table3,5)
```

```
##     interval steps
## 284      835   786
## 285      720   789
## 286      550   794
## 287      900   802
## 288      615   806
```

as we see the interval with the greatest number of steps is 615 with 806 steps, which we will observe in a graph

```
plot(table2$steps ~ table2$interval, type = "l",
     col = "green", xlab = "interval", ylab = "steps")
abline(h = max(table2$steps), v = 615)
```



Imputing missing values

at this point we will impute the NA data and find how many there are in our data frame.

```
list_na <- is.na(activity$steps)
table(list_na)
```

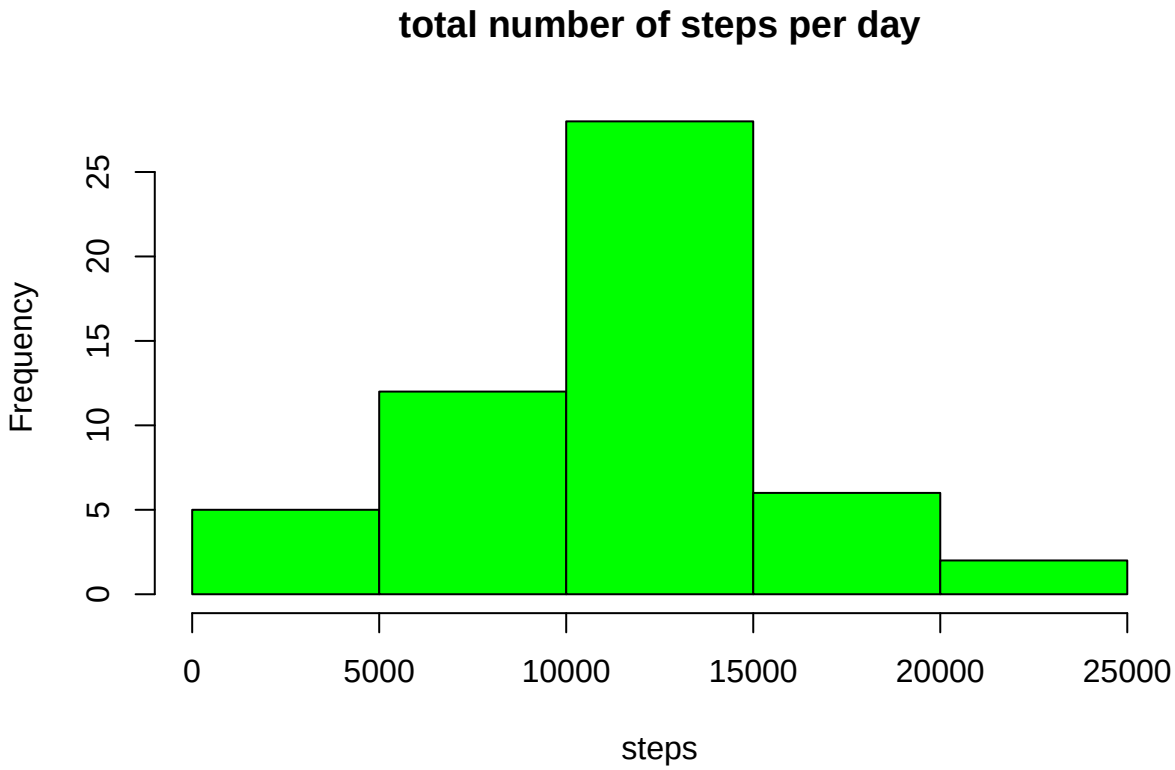
```
## list_na
## FALSE  TRUE
## 15264  2304
```

we have 2304 NA values, so we will replace them from our original data frame with the mean minus the standard deviation of the steps per day.

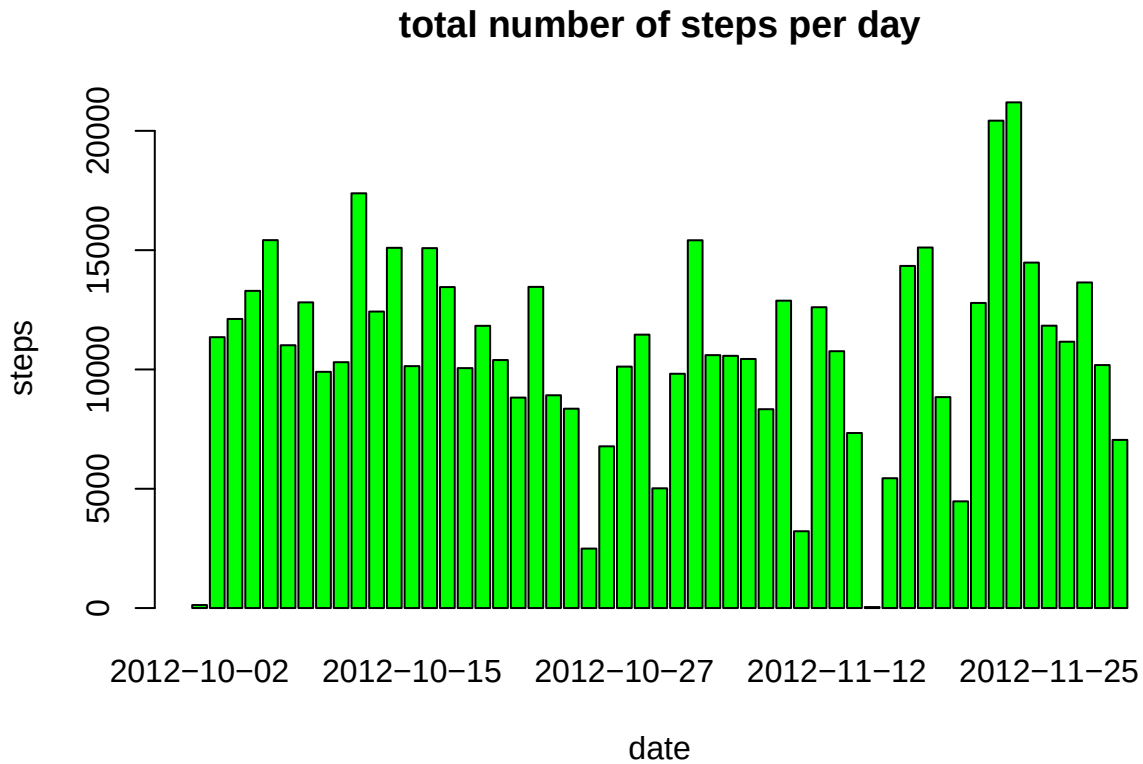
```
t <- mean(df$steps) - sqrt(var(df$steps))
for(i in 1:17568){
        f <- is.na(activity$steps[i])
        if(f == TRUE)
                activity[i,1]<- t }
```

now we will make a histogram and a bar graph to observe the changes obtained with respect to the original data.

```
df1 <- ddply(activity2,.(date),summarise,
             steps = sum(steps))
hist(df1$steps,
     main = "total number of steps per day",
     xlab = "steps", col = "green")
```




```
barplot(df1$steps,names.arg = df1$date, col = "green",
        main = "total number of steps per day",
        xlab = "date", ylab = "steps")
```



In the histogram we can see that the frequency of the steps is mostly between 10,000 and 15,000 steps and seeing the bar graph, the day on which the most steps were taken was 11/19/2012.

now we will compare the mean and median per day of the data without NA values with those with NA values.

```
df$median <- ddply(activity2,.(date),summarise,
                   steps = median(steps))
difmean <- c(mean(dfmm$mean), mean(df$steps))
difmedian <- c(median(dfmm$median$median),
               median(df$median$steps))
print(difmean)
```

```
## [1]       NA 129.7411
```

```
print(difmedian)
```

```
## [1] NA 56
```

we see that the data with NA values have a bias towards them, while the data without these values gives us a numerical value that gives us an idea of how the data is distributed.

Are there differences in activity patterns between weekdays and weekends?

Now we are going to introduce a new variable which we will call type of day, that is, it will have two levels, day of the week and weekend, to observe the steps regarding the intervals in these days.

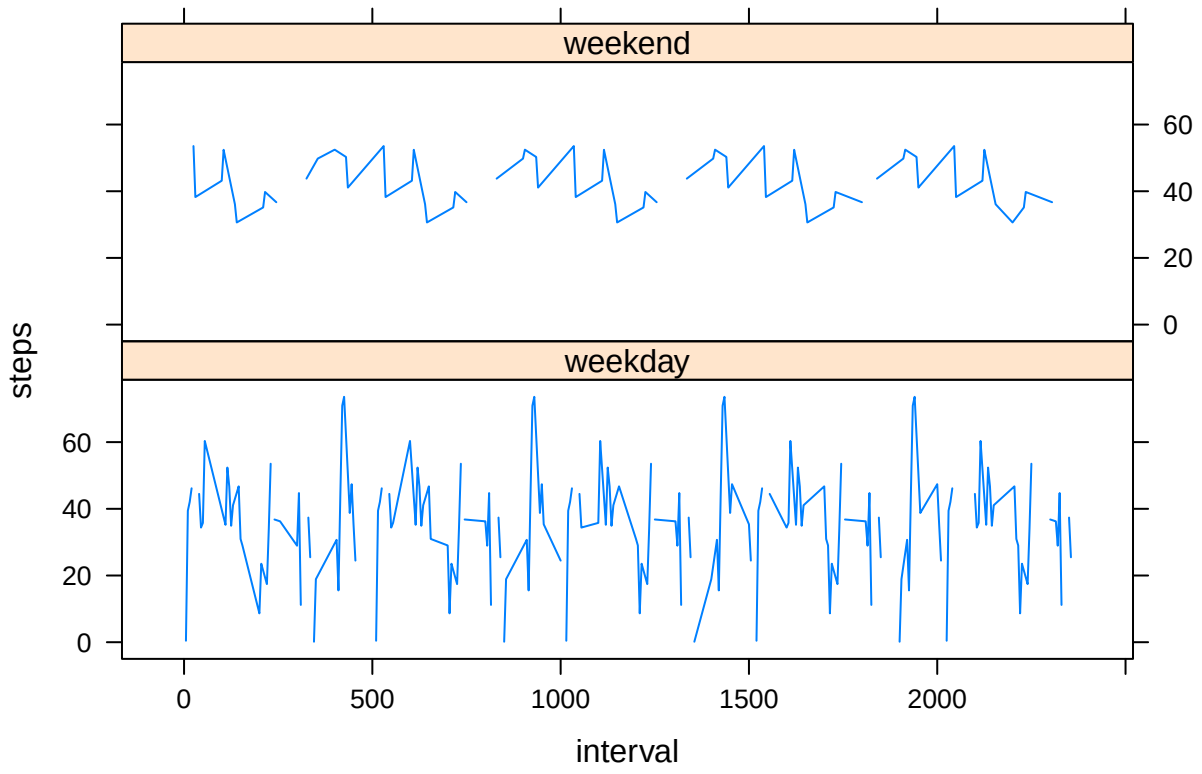
```
dfmm <- dfmm[,-3]
dfmm$date1 <- as.Date(dfmm$date)
dfmm$days <- weekdays(dfmm$date1)
```

```
for(i in 1:61){
        day <- dfmm$days[i]
        if(day == "Saturday" || day == "Sunday")
                dfmm$typeday[i] <- "weekend"
       else{
               dfmm$typeday[i] <- "weekday"}
        }
```

```
xyplot(dfmm$mean ~ table2$interval|dfmm$typeday,
       type = "l",
       layout= c(1,2),
       xlab = "interval", ylab = "steps")
```

```
## Warning in is.na(x) | is.na(y): longer object length is not a multiple of
## shorter object length
```

```
## Warning in is.na(x) | is.na(y): longer object length is not a multiple of
## shorter object length
```


weekend
weekday
steps
interval
0
20
40
60
0
500
1000
1500
2000