Reproducible Rsearch: Peer Assignment 1

Suchir Bhatnagar

9/5/2017

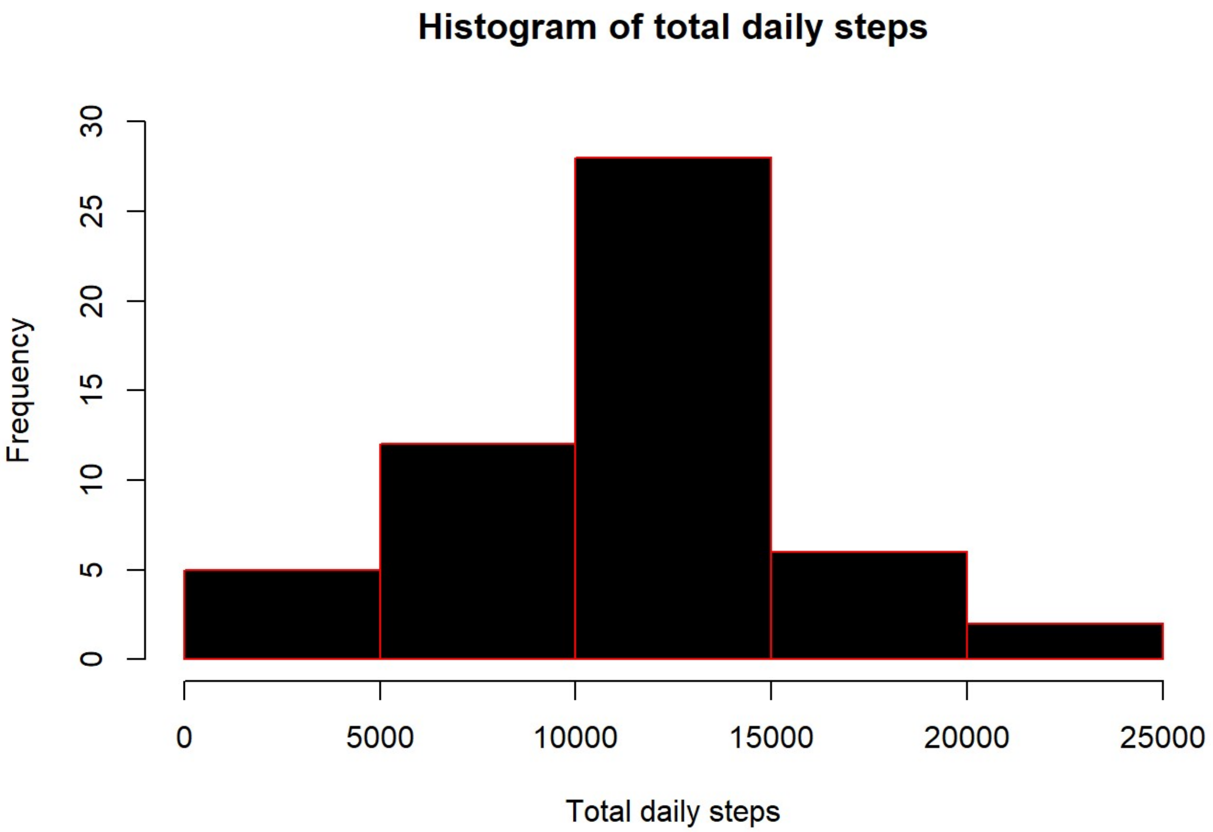
Loading and pre-processing the data

We are first going to download the file, unzip the file and read the data

We would like to know the mean total number of steps taken per day

Let us first calculate the total number of steps per day. Best way to do this would be to aggregate and sum the data on a daily basis.

```r
daily_steps<-group_by(activity_data,date)
total_daily_steps<-summarise(daily_steps,sum(steps))
#png("myplot.png", width=4, height=4, units="in", res=300)
hist(total_daily_steps$`sum(steps)`,main="Histogram of total daily steps",
     xlab="Total daily steps",ylab="Frequency",border="red",col="black",
     ylim=c(0,30))
```



```
#dev.off()
```

The above histogram shows us that the user takes a maximum of 10,000-15,000 steps per day. He has had a few off days when he took zero steps or forgot to turn on the device, on the brighter side he has managed to acheive a peak of 25,000.

```
library(knitr)
mean_total_daily_steps<-mean(total_daily_steps$'sum(steps)',na.rm=TRUE)
median_total_daily_steps<-median(total_daily_steps$'sum(steps)',na.rm=TRUE)
```

He took a mean no. of 10,766 steps per day while the median value hovered around 10765. The two values are almost equal to each other indicating that the user has been following a set routine most of the days. **Have tried to call the inline function here using r variable_name but this wasnot working for me. after a lot of googling failed to find the soluton hence left with no choice but to hardcode the values.**

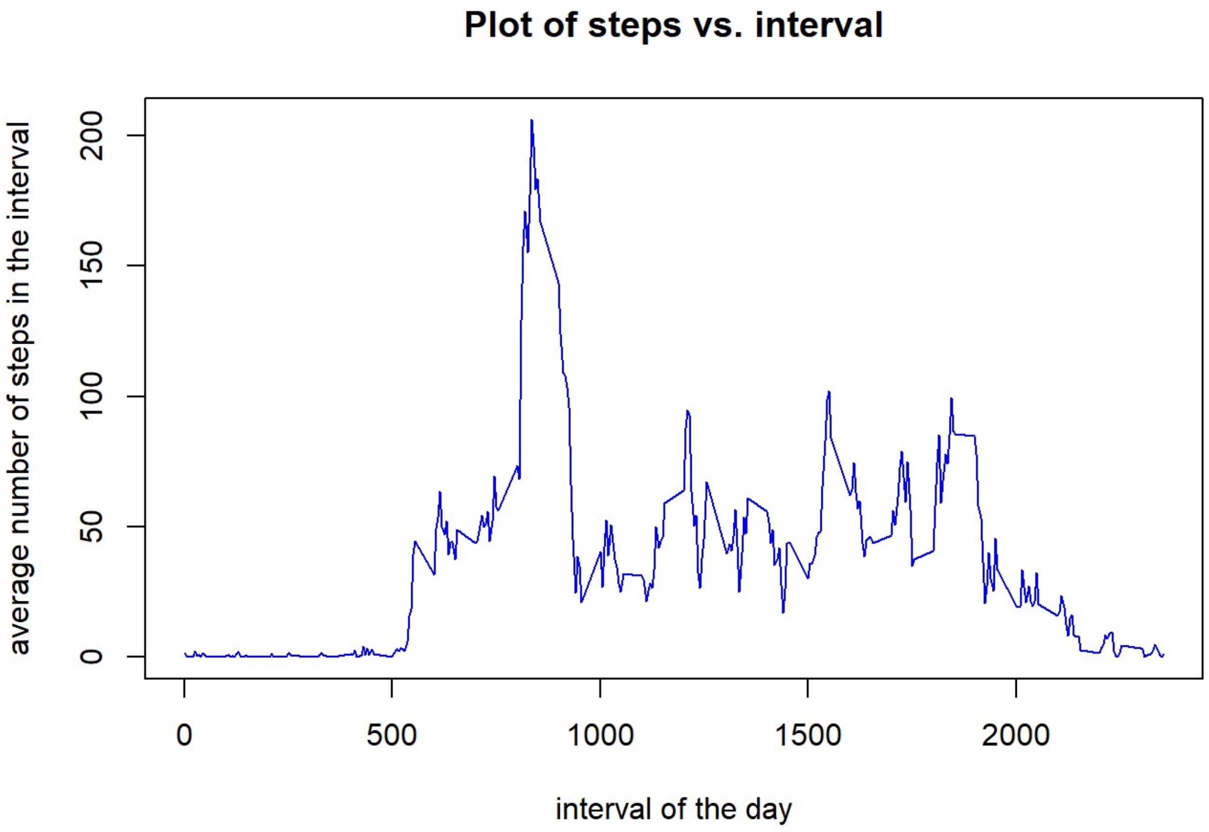
Let us understand the average daily pattern of the user

we would like to plot the average number of steps for a given time interval across all days

```
average_daily_steps<-group_by(activity_data,interval)
average_steps<-summarise(average_daily_steps,mean(steps,na.rm=TRUE))
which.max(average_steps$`mean(steps, na.rm = TRUE)`)
```

```
## [1] 104
```

```
plot(average_steps,type="l",ylab="average number of steps in the interval",xlab="interval of the day",main="Plot of steps vs. interval",col="blue")
```



The above shows the plot of the average number of steps during the day. As expected the graph shows a user sleeping for the early parts of the day and is most active during the early half of the day, hitting a peak at: 104th interval.

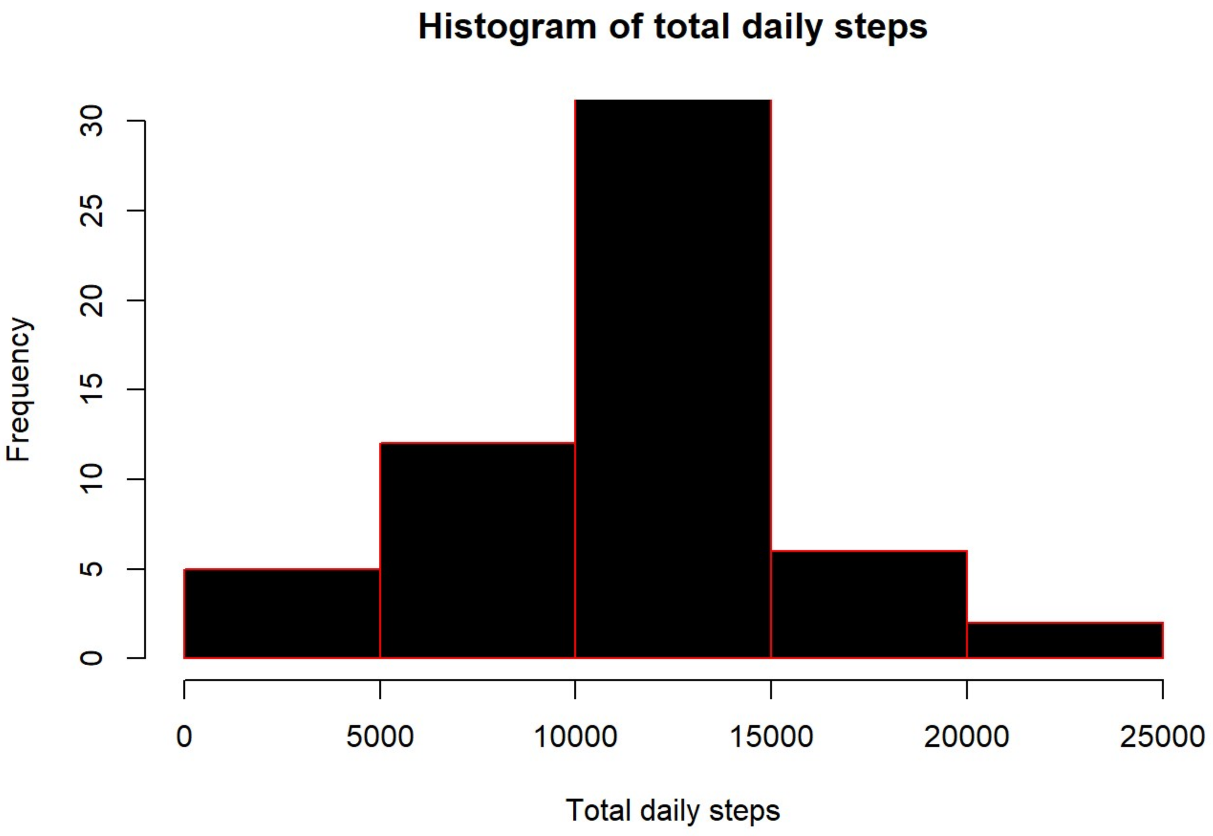
Imputing Missing Values

In this exercise we will first find out what are the number of missing values, how can we replace these - for this we will replace it with the mean of that interval.

```
missing_data<-which(is.na(activity_data$steps))
len_missing_data<-length(missing_data)
names(average_steps)=c("interval","steps")

for (i in 1:len_missing_data) {
index_1<-missing_data[i]
missing_interval<-activity_data[index_1,3]
index_2<-which(average_steps$interval==missing_interval)
average_for_interval<-average_steps[index_2,2]
activity_data[index_1,1]=average_for_interval
}

daily_steps<-group_by(activity_data,date)
total_daily_steps<-summarise(daily_steps,sum(steps))
hist(total_daily_steps$`sum(steps)`,main="Histogram of total daily steps",
     xlab="Total daily steps",ylab="Frequency",border="red",col="black",
     ylim=c(0,30))
```



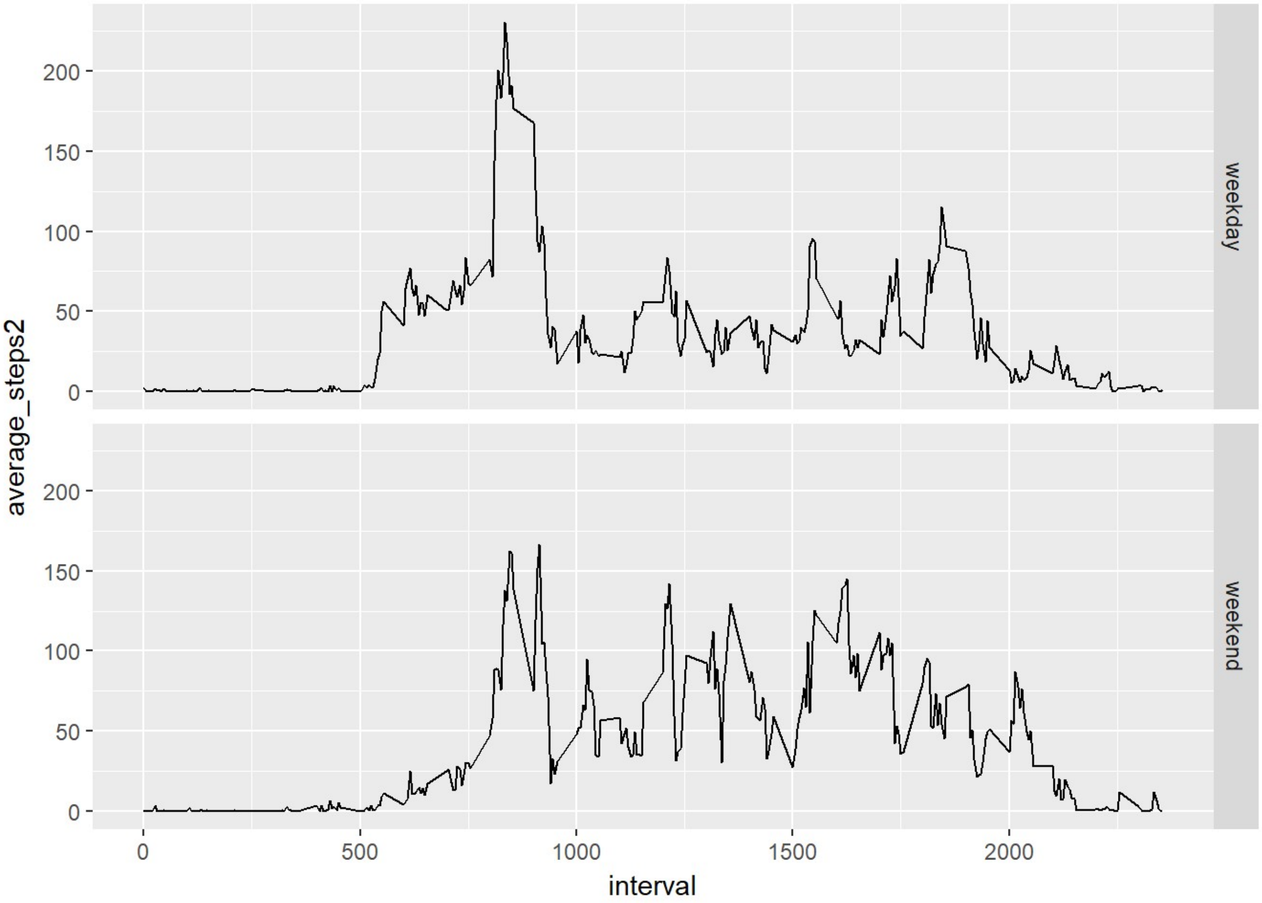
```
mean_total_daily_steps<-mean(total_daily_steps$'sum(steps)',na.rm=TRUE)
median_total_daily_steps<-median(total_daily_steps$'sum(steps)',na.rm=TRUE)
```

This shows that there are 2304 missing values in the data set. After imputing the missing values we find that there is not much difference in the activity level across the days, showing a regular activity pattern by the user. Also the mean and median of total daily steps = 10,766 compared to 10,766 in the first scenario without imputing the values.

Are there any differences in activities between weekdays and weekends

Let us first identify the weekdays and the weekends, we will use the code below for that.

```
activity_data$date<-as.Date(activity_data$date)
activity_data$day_of_week<-weekdays(activity_data$date)
sat_index<-which(activity_data$day_of_week =="Saturday")
sun_index<-which(activity_data$day_of_week=="Sunday")
weekend_index<-rbind(sat_index,sun_index)
activity_data$type_of_day<-"weekday"
activity_data[weekend_index,"type_of_day"]<-"weekend"
activity_data$type_of_day<-as.factor(activity_data$type_of_day)
activity_data_gr_week<-group_by(activity_data,interval,type_of_day)
activity_data_summary<-summarise(activity_data_gr_week,mean(steps,na.rm=TRUE))
names(activity_data_summary)[3]<-"average_steps2"
p<-ggplot(activity_data_summary,aes(interval,average_steps2))+geom_line()
p+facet_grid(type_of_day~.)
```



Now that we have separated the two, we can take an average of the number of steps across the various intervals during the weekdays and the weekends. We see that while the weekdays have a peak performance the weekends tend to be more subdued with a slightly flat average number of steps during the day.