

Activity

Praveen

14 August 2015

```
library(ggplot2)
library(scales)
library(Hmisc)
```

```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: splines
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##      format.pval, round.POSIXt, trunc.POSIXt, units
```

```
library(timeDate)
library(lattice)
```

```
##Load the data
```

```
##Process/transform the data (if necessary) into a format suitable for your analysis
```

```
data1<-read.csv("/home/praven/Dropbox/R Programming/activity.csv")
```

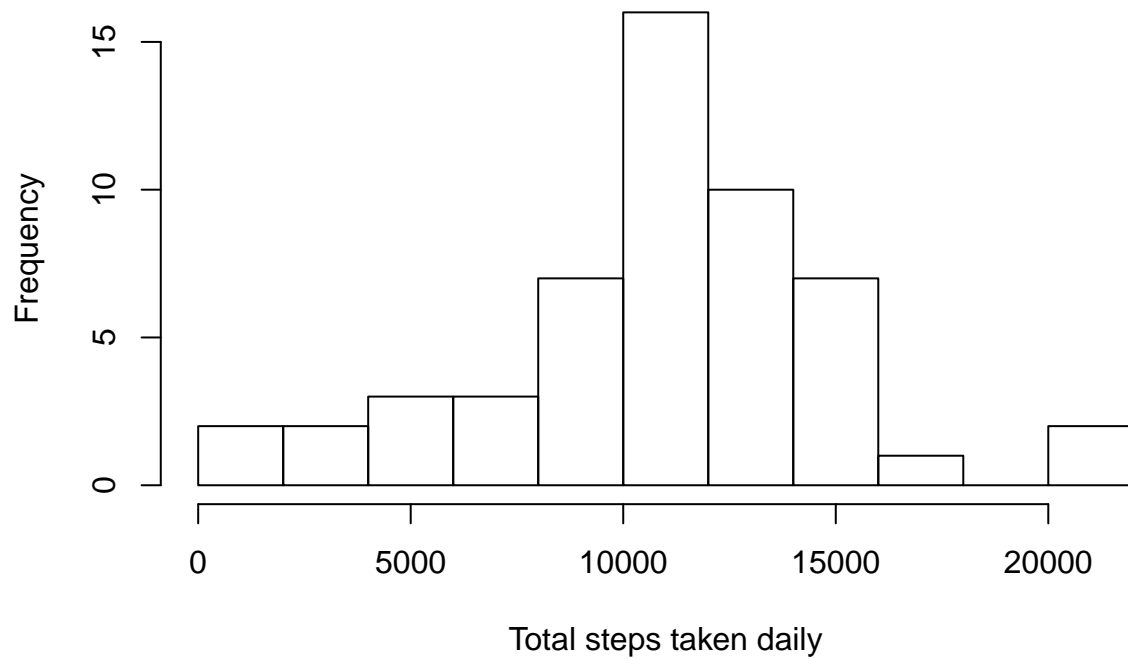
```
##Calculate the total number of steps taken per day
```

```
total_steps<-aggregate(steps~date,data=data1,sum,na.action = na.omit)
```

```
##Make a histogram of the total number of steps taken each day
```

```
hist(total_steps$steps,breaks=10,xlab="Total steps taken daily",main="Histogram of total steps taken ea
```

Histogram of total steps taken each day



```
##Calculate and report the mean and median of the total number of steps taken per day
```

```
mean(total_steps$steps)
```

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

```
median(total_steps$steps)
```

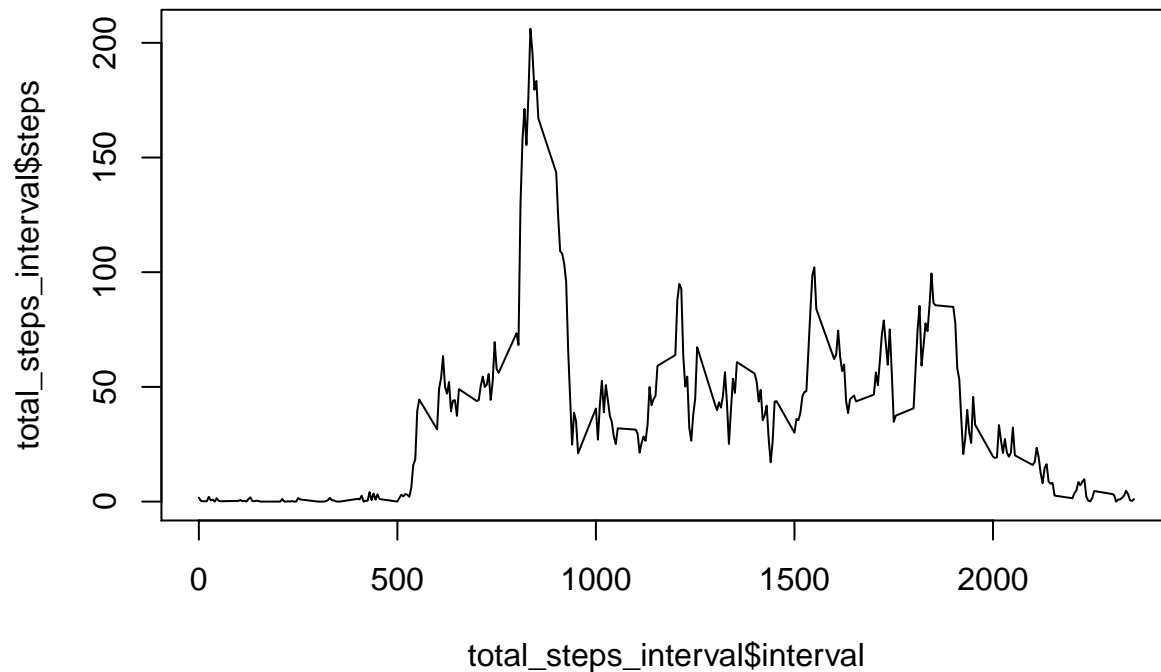
```
## [1] 10765
```

```
##Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of
```

```
total_steps_interval<-aggregate(steps~interval,data=data1,FUN=mean,na.action = na.omit)
```

```
##Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of
```

```
plot(total_steps_interval$interval,total_steps_interval$steps,type="l")
```



```
total_steps_interval[which.max(total_steps_interval$steps), ]
```

```
##      interval      steps
## 104         835 206.1698
```

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

```
sum(is.na(data1))
```

```
## [1] 2304
```

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

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

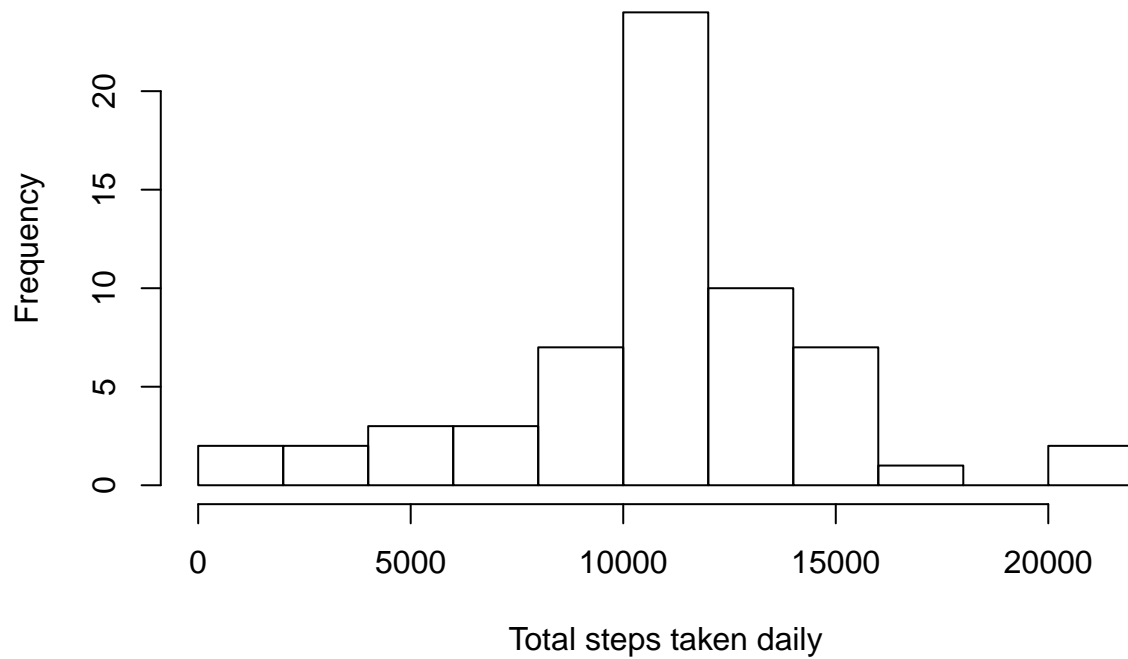
```
imputed_data <- transform(data1, steps = ifelse(is.na(data1$steps), total_steps_interval$steps[match(da
```

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

```
total_steps1<-aggregate(steps~date,data=imputed_data,sum,na.action = na.omit)
```

```
hist(total_steps1$steps,breaks=10,xlab="Total steps taken daily",main="Histogram of total steps taken e
```

Histogram of total steps taken each day



```
mean(total_steps1$steps)
```

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

```
median(total_steps1$steps)
```

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

```
##Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether
```

```
imputed_data$weekdays<-isWeekday(imputed_data$date,wday=1:5)
for(i in 1:nrow(imputed_data)) {
  if(isWeekday(as.Date(imputed_data[i,2]),wday=1:5)) imputed_data[i,4]<-1 else imputed_data[i,4]<-0
}
```

```
##Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and
```

```
imputed_data_interval<-aggregate(steps~interval+weekdays,data=imputed_data,FUN=mean)
```

```
xyplot(steps~interval |weekdays,data=imputed_data_interval,type='l',main="Mean steps per 5 minute interval by weekday")
```

Mean steps per 5 minute interval for weekend and weekdays

