Reproducible Research: Peer Assessment 1

library(dplyr) library(lattice) ##Setting Working Directory

```
setwd("C:/Users/trint/Desktop")
df = read.csv("activity.csv")
head(df)
```

```
##   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
```

```
library(dplyr)
```

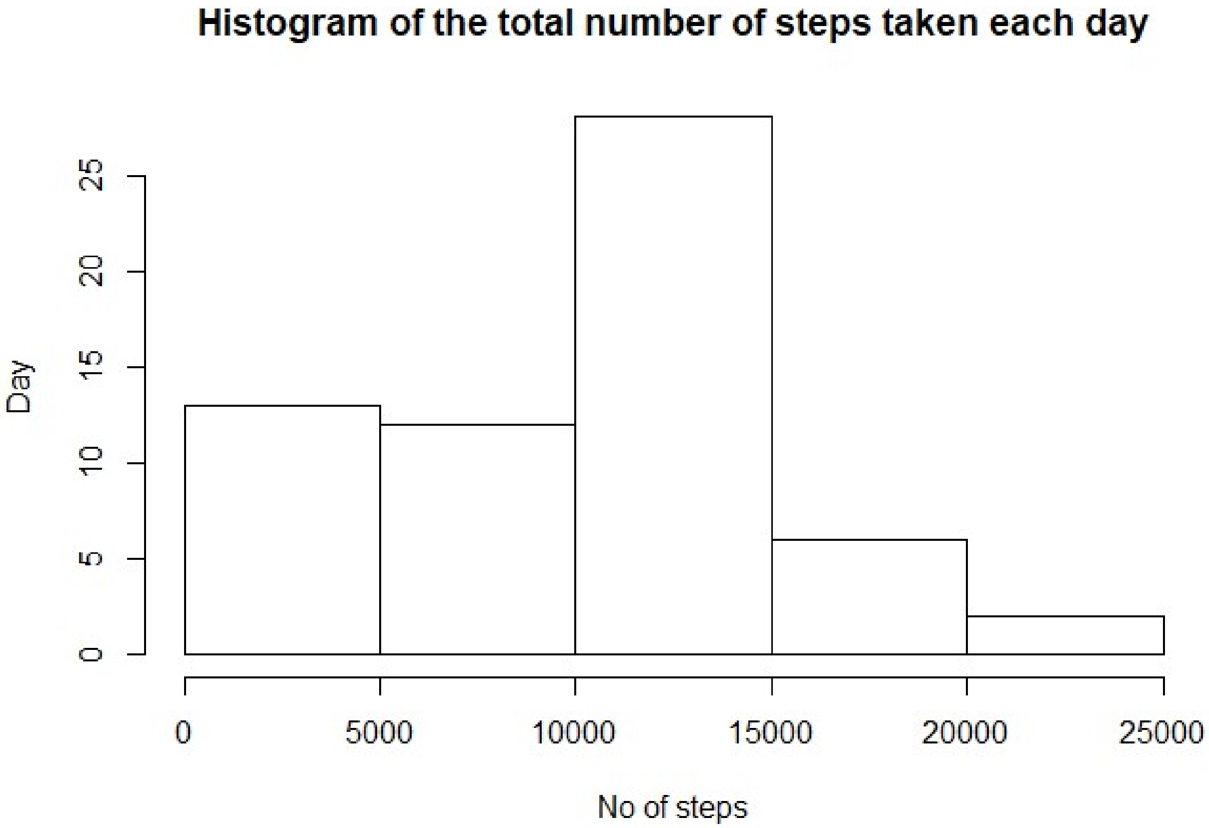
```
##
## Attaching package: 'dplyr'
```

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

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

```
#Calculate the total number of steps taken per day
total = df %>% group_by (date) %>% summarise (total = sum(steps,na.rm =T))

#Difference between a histogram and a barplot,
hist(total$total, xlab="No of steps", ylab="Day", main="Histogram of the total number
of steps taken each day")
```



```
#Mean and median of the total number of steps taken per day
mean = total %>% summarise (mean = mean(total))
mean
```

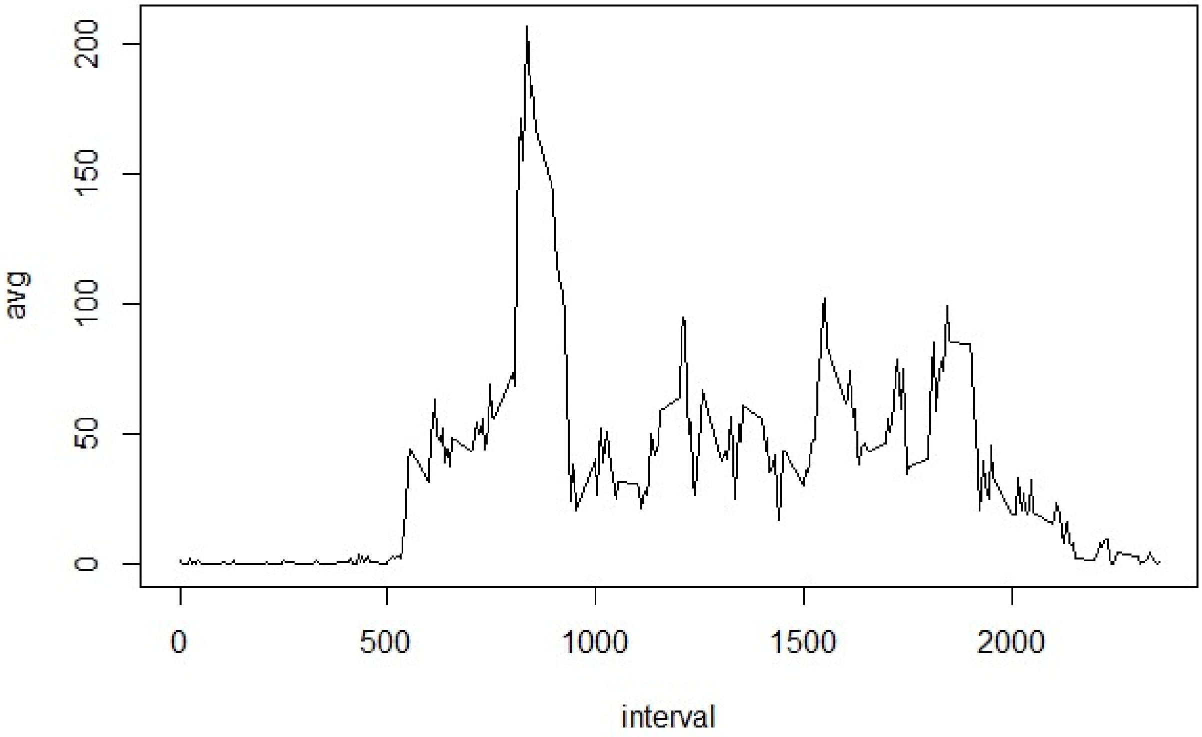
```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1 9354.
```

```
median = median(as.numeric(total$total))
median
```

```
## [1] 10395
```

#time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
ts_interval = df %>% group_by (interval) %>% summarise (avg = mean(steps,na.rm =T))
with(ts_interval, plot(interval, avg, type = "l"))
```



```
ts_interval = as.data.frame(ts_interval)
max_step = max(ts_interval[,"avg"])
ts_interval %>% filter (avg == max_step)
```

```
##   interval      avg
## 1      835 206.1698
```

```
# total number of missing values in the dataset
length(which(is.na(df$steps) == T))
```

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

#missing values in the datasetimputed with mean for that day

```
total = df %>% group_by (date) %>% summarise (total = sum(steps,na.rm =T))
total = as.data.frame(total)
count = df %>% group_by (date) %>% tally ()
count = as.data.frame(count)
total = merge(total,count,by="date")
total$mean = ifelse(total$total ==0,0, total$total/total$n)
total = total[,c("date","mean")]
df = merge(df,total, by="date")
df2 = df %>% group_by(date) %>% mutate (steps = ifelse(is.na(steps), mean, steps))
df2 = as.data.frame(df2)
df2 = df2[,-c(4)]
head(df2)
```

```
##         date steps interval
## 1 2012-10-01     0        0
## 2 2012-10-01     0        5
## 3 2012-10-01     0       10
## 4 2012-10-01     0       15
## 5 2012-10-01     0       20
## 6 2012-10-01     0       25
```

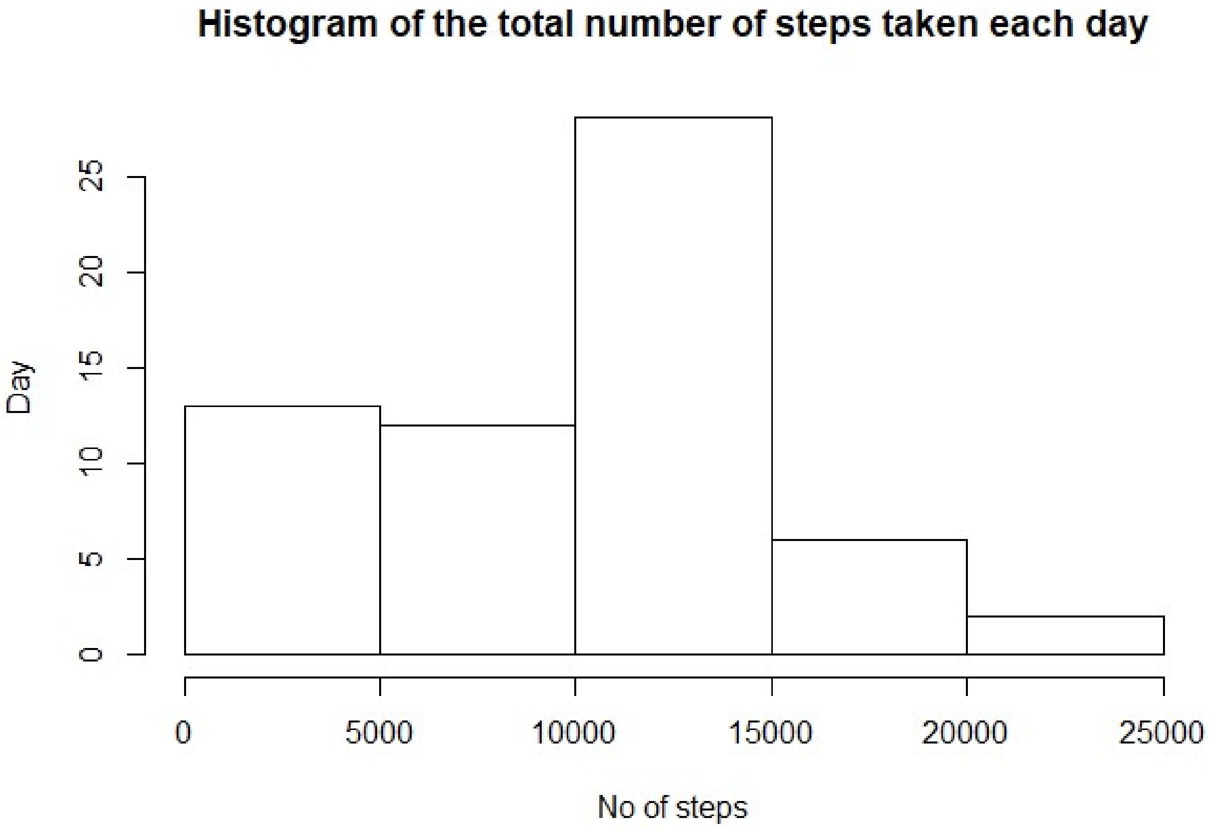
#histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. #Calculate the total number of steps taken per day

```
total = df2 %>% group_by (date) %>% summarise (total = sum(steps,na.rm =T))
total
```

```
## # A tibble: 61 x 2
##    date       total
##    <fct>      <dbl>
##  1 2012-10-01     0
##  2 2012-10-02   126
##  3 2012-10-03 11352
##  4 2012-10-04 12116
##  5 2012-10-05 13294
##  6 2012-10-06 15420
##  7 2012-10-07 11015
##  8 2012-10-08     0
##  9 2012-10-09 12811
## 10 2012-10-10  9900
## # ... with 51 more rows
```

#Difference between a histogram and a barplot,

```
hist(total$total, xlab="No of steps", ylab="Day", main="Histogram of the total number
of steps taken each day")
```



```
#Mean and median of the total number of steps taken per day
mean2 = total %>% summarise (mean = mean(total))
mean2
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1 9354.
```

```
mean
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1 9354.
```

```
median2 = median(as.numeric(total$total))
median2
```

```
## [1] 10395
```

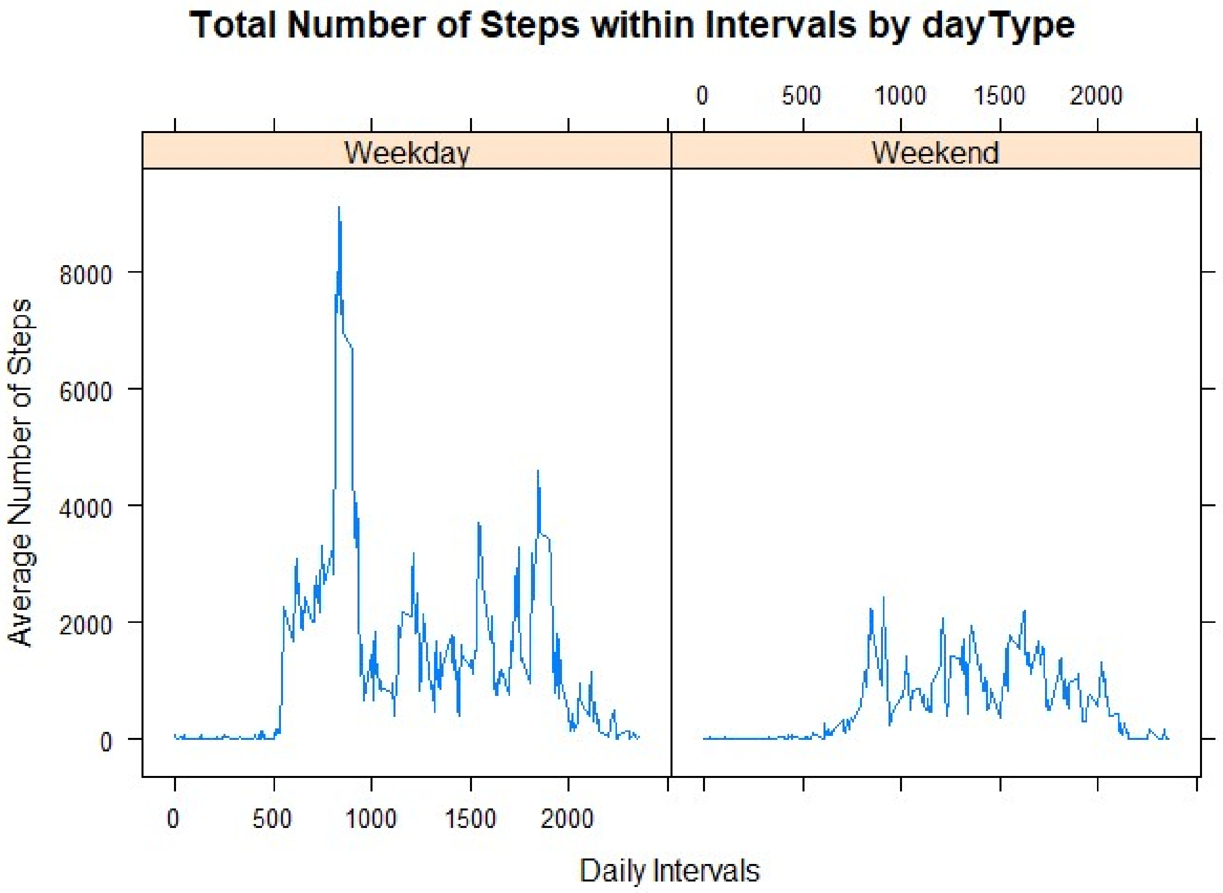
```
median
```

```
## [1] 10395
```

#no diff for mean and median

#Are there differences in activity patterns between weekdays and weekends?

```
library(dplyr)
df2$date = as.Date(df2$date)
df2 = df2 %>% mutate(Indicator = ifelse(weekdays(df2$date)=="Saturday" | weekdays(df2
$date)=="Sunday", "Weekend", "Weekday"))
library(dplyr)
ts = df2 %>% group_by(Indicator, interval) %>% summarize(total=sum(steps))
library(lattice)
with(ts, xyplot(total ~ interval | Indicator,
            type = "l",
            main = "Total Number of Steps within Intervals by dayType",
            xlab = "Daily Intervals",
            ylab = "Average Number of Steps"))
```


Total Number of Steps within Intervals by dayType
Weekday
Weekend
Average Number of Steps
0
2000
4000
6000
8000
Daily Intervals
0
500
1000
1500
2000