Week 2 Reproducible research assignment

First, we read in the csv file and perform some pre-processing

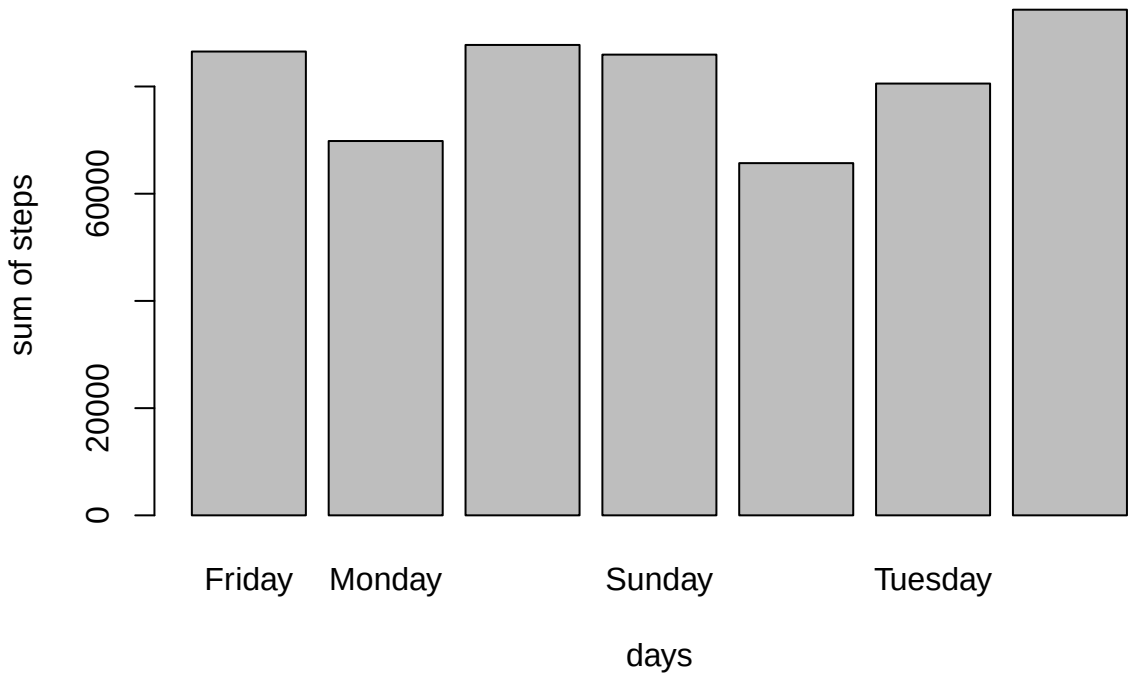
```
library("data.table")
library(ggplot2)
```

```
##read in the data and some pre pocessing

myfile <- read.csv("C:/Users/CVSSP/Desktop/R Programming/activity.csv")
Date <- as.Date(myfile$date)
myfile$date= NULL
myfile$Date <- Date
myfile$day <- weekdays(Date)
```

Next, we can plot the total number of steps each day as follows:

```
mydata= aggregate(x= myfile$steps, by= list(myfile$day), FUN= sum, na.rm = TRUE)
names= mydata$Group.1
barplot(mydata$x,main = "Average steps per weekday", names.arg = names, xlab = "days", ylab = "sum of st
```



Next we find the mean and median of the total steps per day as follows:

```
mean(mydata$x)
```

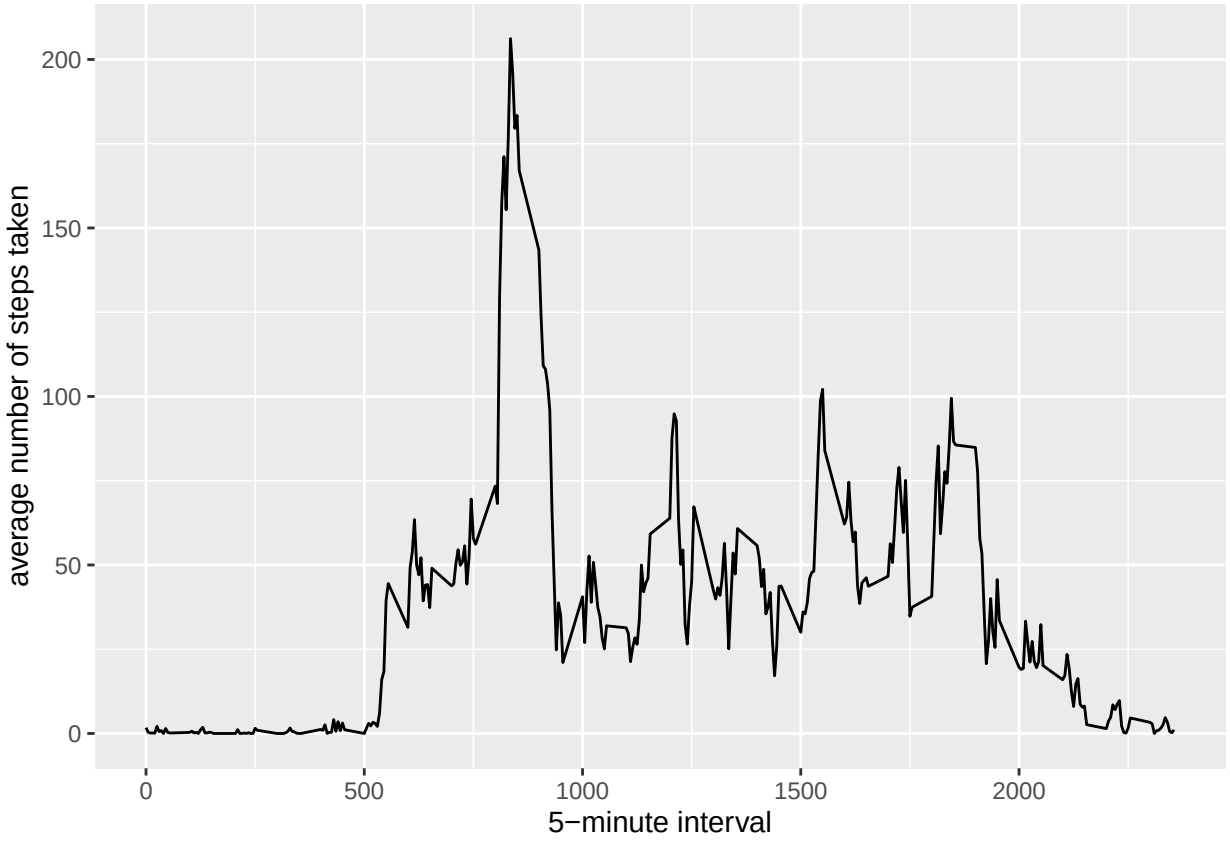
```
## [1] 81515.43
```

```
median(mydata$x)
```

```
## [1] 85944
```

To plot a time series chart of all the different time intervals along with the interval with the max steps

```
library(ggplot2)
averages <- aggregate(x=list(steps=myfile$steps), by=list(interval=myfile$interval),
                      FUN=mean, na.rm=TRUE)
ggplot(data=averages, aes(x=interval, y=steps)) +
    geom_line() +
    xlab("5-minute interval") +
    ylab("average number of steps taken")
```




```
averages[which.max(averages$steps),]  ## to find the interval with the highest number of steps
```

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

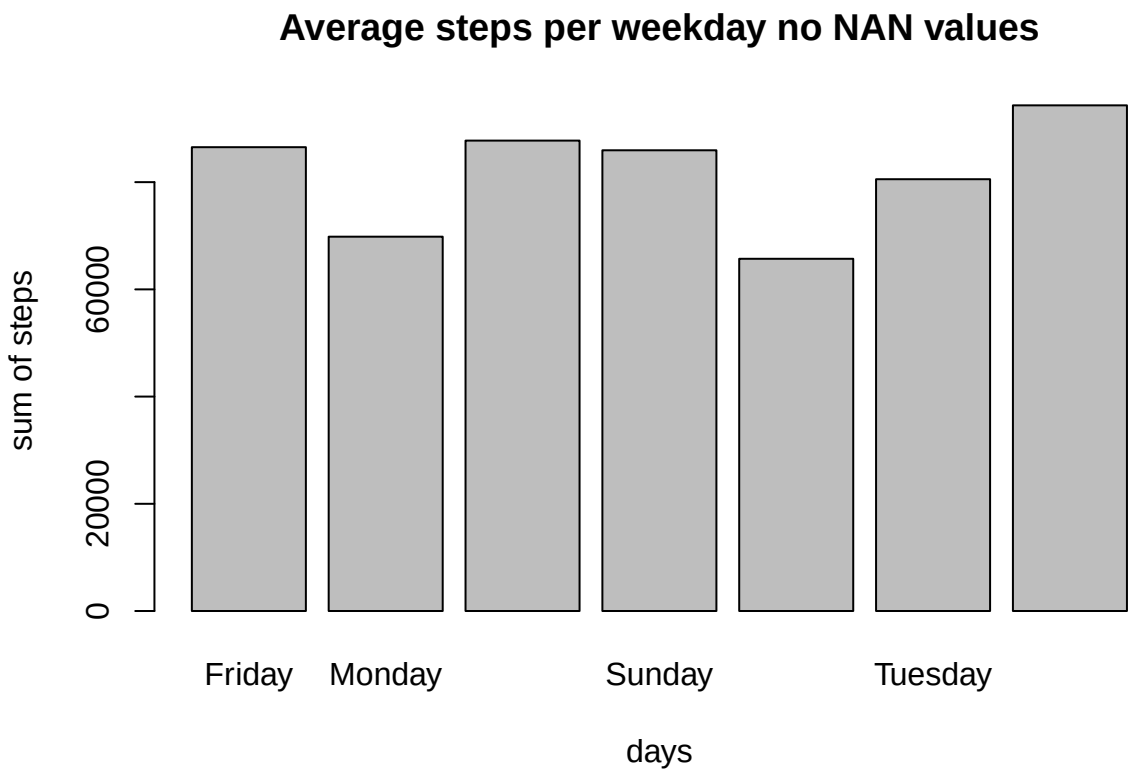
Now we will consider the NA values and the effect they have on our analysis. To count the number of NA values:

```r
sum(is.na(myfile))
```

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

Next we devise a way to fill in the missing data

```r
myfile$steps[is.na(myfile$steps)]<-mean(mydata$x,na.rm=TRUE)
barplot(mydata$x,main = "Average steps per weekday no NAN values", names.arg = names, xlab = "days", yla
```




```r
mean(mydata$x)
```

```
## [1] 81515.43
```

```r
median(mydata$x)
```

```
## [1] 85944
```

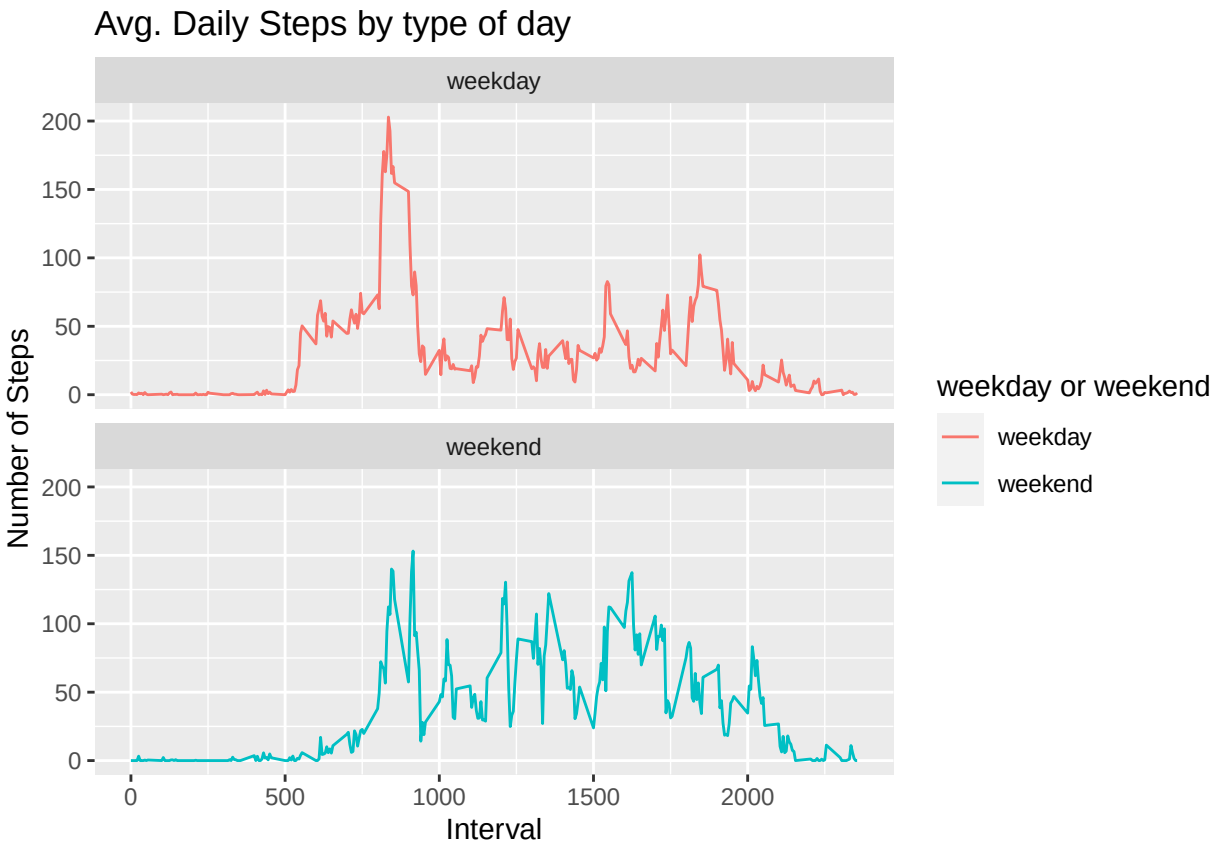
There does not seem to be a noticeable change to the mean or median values after this simple imputation of the data. The estimates of the total daily number of steps naturally increased.

Finally, we compare activity levels on weekends vs weekdays

```
q4database <- data.table::fread(input = "C:/Users/CVSSP/Desktop/R Programming/activity.csv")
q4database[, date := as.POSIXct(date, format = "%Y-%m-%d")]
q4database[, `Day of Week`:= weekdays(x = date)]
q4database[grepl(pattern = "Monday|Tuesday|Wednesday|Thursday|Friday", x = `Day of Week`), "weekday or w
q4database[grepl(pattern = "Saturday|Sunday", x = `Day of Week`), "weekday or weekend"] <- "weekend"
q4database[, `weekday or weekend` := as.factor(`weekday or weekend`)]

q4database[is.na(steps), "steps"] <- q4database[, c(lapply(.SD, median, na.rm = TRUE)), .SDcols = c("ste
IntervalDT <- q4database[, c(lapply(.SD, mean, na.rm = TRUE)), .SDcols = c("steps"), by = .(interval, `w

ggplot(IntervalDT , aes(x = interval , y = steps, color=`weekday or weekend`)) + geom_line() + labs(titl
```



Overall, there is a noticable difference between activity on weekdays and weekends where till 08:00 AM on weekends the activity is considerably less. However, through the rest of the day activity on average is more than that on weekdays.