Peer 1 Assignment

Preliminaries

```
library(knitr)
opts_chunk$set(echo = TRUE, results = "hold")

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

Loading and preprocessing the data

The data in the *repdata-data-activity.zip* comes from a step counter that gathers personal data at 5 minute intervals throughout the day. The data was collected in October and Novemeber of 2012.

1.Loading the data

```
sdat <- read.csv(unz('repdata-data-activity.zip', 'activity.csv'), header =TRUE)
```

2. Cleaning the Data

```
sdat$date <- as.Date(sdat$date, format = "%Y-%m-%d")
sdat$interval <- as.factor(sdat$interval)
```

Number of steps taken in a day

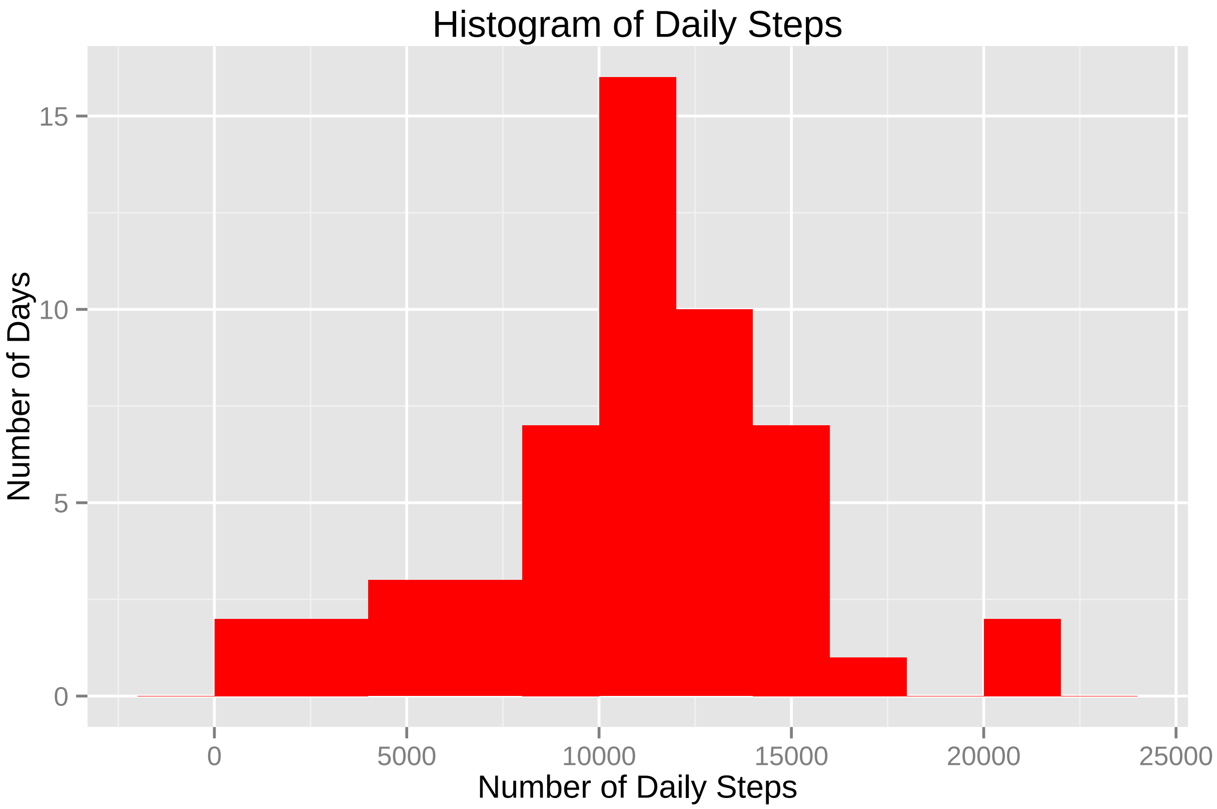
1. We can aggregate the number of steps taken per day.

```
daily_steps <- aggregate(steps ~ date, sdat, sum)
colnames(daily_steps) <- c("date", "steps")
head(daily_steps)
```

```
##         date steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

2. Histogram of Number of steps taken per day.

```
ggplot(daily_steps, aes(x = steps)) +
  geom_histogram(fill = "red", binwidth = 2000) +
  labs(title = "Histogram of Daily Steps",
       x = "Number of Daily Steps", y = "Number of Days")
```



3. The mean and median of steps taken per day.

```
steps_mean   <- round(mean(daily_steps$steps, na.rm=TRUE))
steps_median <- median(daily_steps$steps, na.rm=TRUE)
```

```
## [1] 10766
## [1] 10765
```

What is the average daily activity pattern?

1.

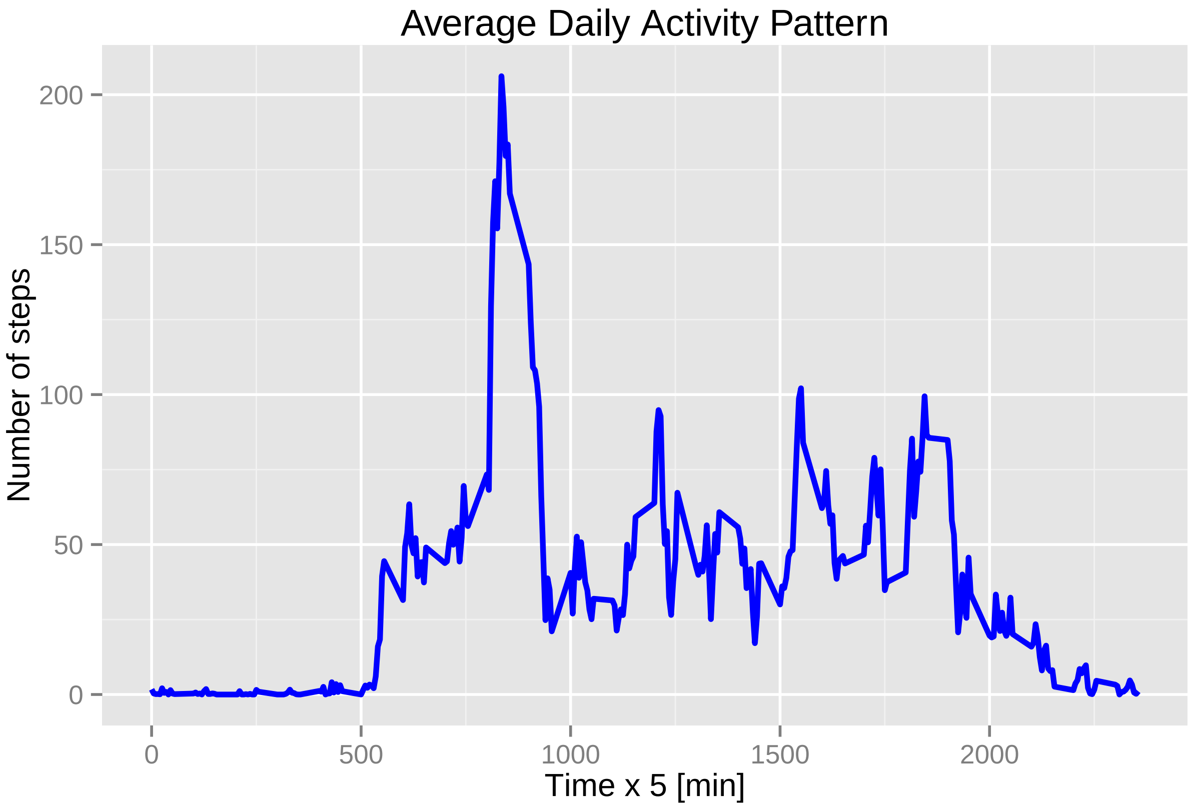
```
dt_steps <- aggregate(x=list(steps=sdat$steps), by = list(interval = sdat$interval), FUN=mean, na.rm=TRU

head(dt_steps)
dt_steps$interval <-
        as.integer(levels(dt_steps$interval)[dt_steps$interval])
colnames(dt_steps) <- c("interval", "steps")
```

```
##   interval     steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

We can now plot the number of steps taken in a 5 minute interval.

```
ggplot(dt_steps, aes(x=interval, y=steps)) +
        geom_line(color="blue", size=1) +
        labs(title="Average Daily Activity Pattern", x="Time x 5 [min]", y="Number of steps")
```



2. The maximum number of steps can be readily calculated.

```
max_steps <- dt_steps[which.max(dt_steps$steps),]
```

```
## [1] 206.1698
```

Imputting missing values

1. The number of missing *NA* values is simply:

```
na_steps <- sum(is.na(sdat$steps))
```

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

2. Fill the data with the **fill_na** function which utilizes **mapply**. A filled set is returned that replaces all *NA* values with the average value for that interval calculated in *dt_steps*.

```
fill_na <- function(steps, interval){
  filled <- NA
  if (!is.na(steps))
    filled <- c(steps)
  else
    filled <- (dt_steps[dt_steps$interval==interval, "steps"])
  return(filled)
}
```

3. A new data set *sData* has *NA* replaced with the averages.

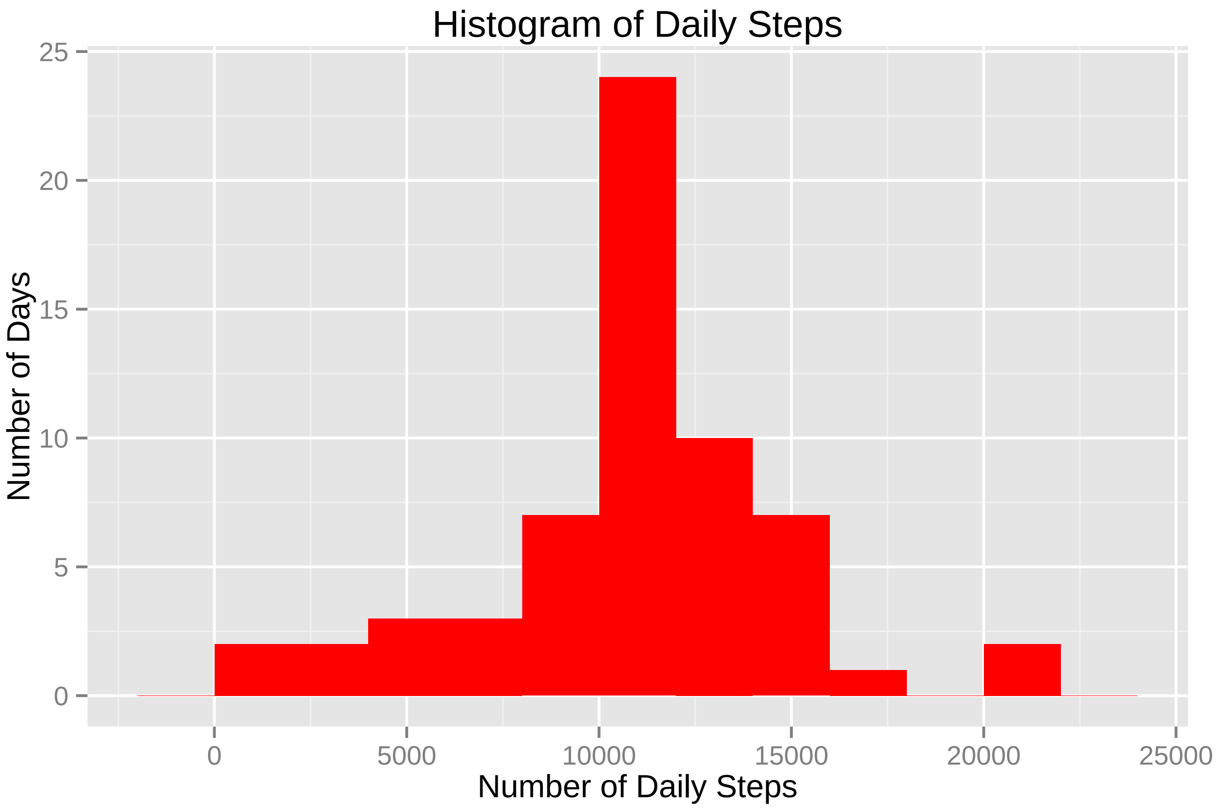
```
sData <- sdat
sData$steps <-mapply(fill_na, sData$steps, sData$interval)
```

```
head(sData)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

4. Plot new histogram with filled-in values.

```
sData_sum <- aggregate(steps ~ date, sData, sum)
ggplot(sData_sum, aes(x = steps)) +
  geom_histogram(fill = "red", binwidth = 2000) +
  labs(title = "Histogram of Daily Steps",
       x = "Number of Daily Steps", y = "Number of Days")
```



```r
sD <- tapply(sData$steps, sData$date, FUN=sum)
mean(sD)
median(sD)
```

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

Weekdays vs. Weekends

1. We now write a function that determines wheter or not a given day is a weekday or weekend.

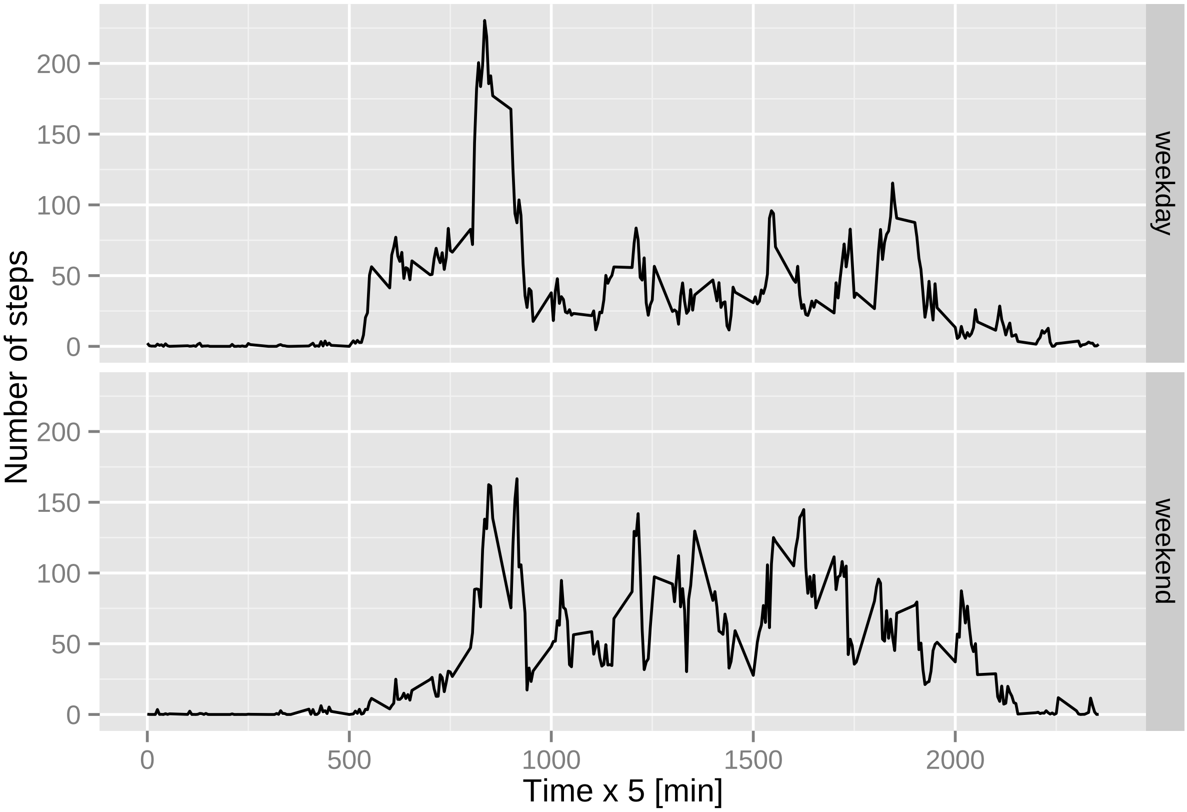
```r
whatday <- function(date) {
    day <- weekdays(date)
    if (day %in% c("Sunday", "Saturday"))
        return("weekend")
    else
        return("weekday")
}
```

2. Append either weekend or weekday to the data frame.

```r
sData$date <- as.Date(sData$date)
sData$day <- sapply(sData$date, FUN=whatday)
```

3. Plot two histograms with a filter on weekday or weekend.

```
averages <- aggregate(steps ~ interval + day, data=sData, mean)
averages$interval <-
        as.integer(levels(averages$interval)[averages$interval])
ggplot(averages, aes(interval, steps)) + geom_line() + facet_grid(day ~ .) +
    xlab("Time x 5 [min]") + ylab("Number of steps")
```


weekday
weekend
Number of steps
Time x 5 [min]