This analysis is for the first *Peer Assessment* project in JHU's online “Reproducible Research” course. The project treats data measured from personal movement monitoring devices. Specifically, the data contains step counts, over 5-minute intervals, recorded between October 1 and November 30, 2012.

Loading and Preprocessing

First the data must be downloaded if not present in the working directory, unzipped and loaded. We want the date variable to be a date class instead of a character, so we need to convert.

```r
## Obtain the data file if not already present
if (!file.exists("activity.csv")) {
  download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Fa
              "repdata-data-activity.zip",method="wget")
  unzip("repdata-data-activity.zip")}

## Load data
rawData <- read.csv("activity.csv",stringsAsFactors = FALSE)

## Convert date
rawData$date <- as.Date(rawData$date)
```

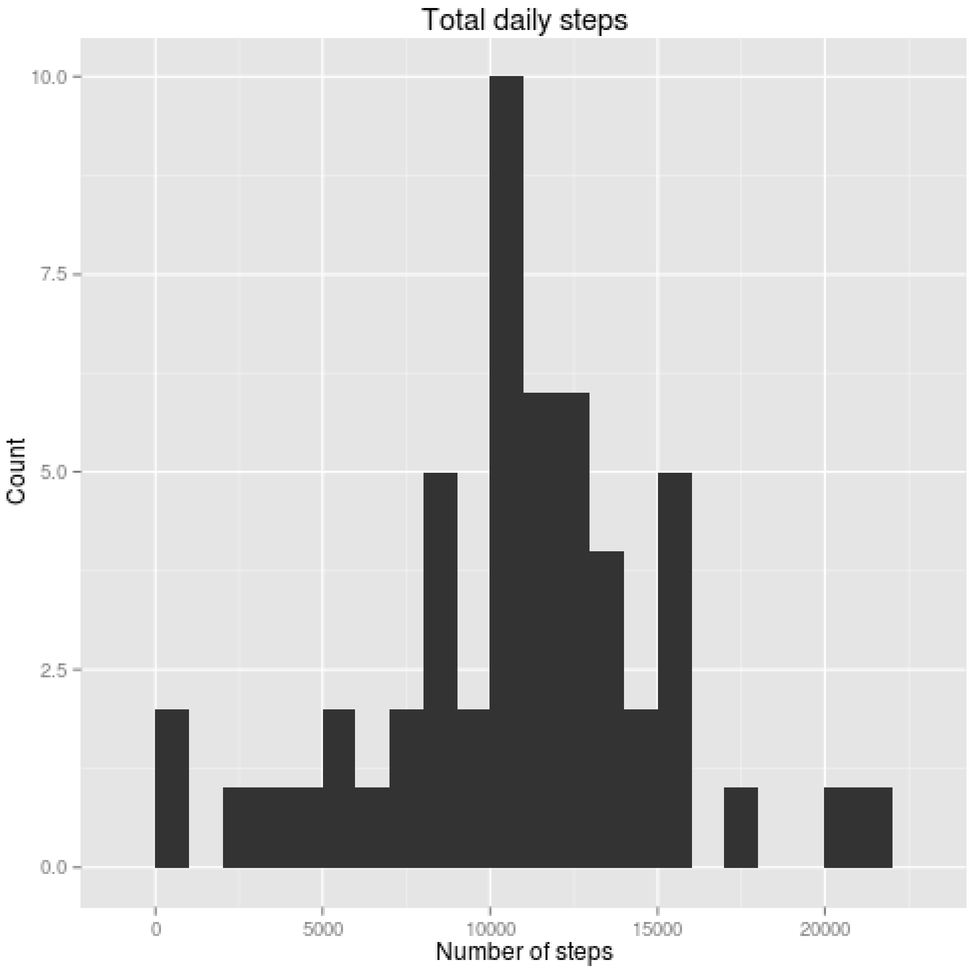
Total daily steps

To find the total daily step counts, we need to remove the unrecorded intervals, and then use the `aggregate` function:

```r
hasSteps <- rawData[!is.na(rawData$steps),]     # Remove NA's
dailySteps <- aggregate(steps~date, hasSteps,sum)   # Sum the steps for
```

Let's use `ggplot2` to plot a histogram. The counts clearly cluster around 10,000.

```r
library(ggplot2)
dailyHist <- qplot(dailySteps$steps, binwidth = 1000, geom="histogram")
dailyHist <- dailyHist + labs(title="Total daily steps")
dailyHist <- dailyHist + labs(x="Number of steps", y="Count")
print(dailyHist)
```



Using the `summary` function, we see that the mean is 10766.19 and the median is 10765. Very close to each other.

```
summary(dailySteps$steps,digits=7)
```

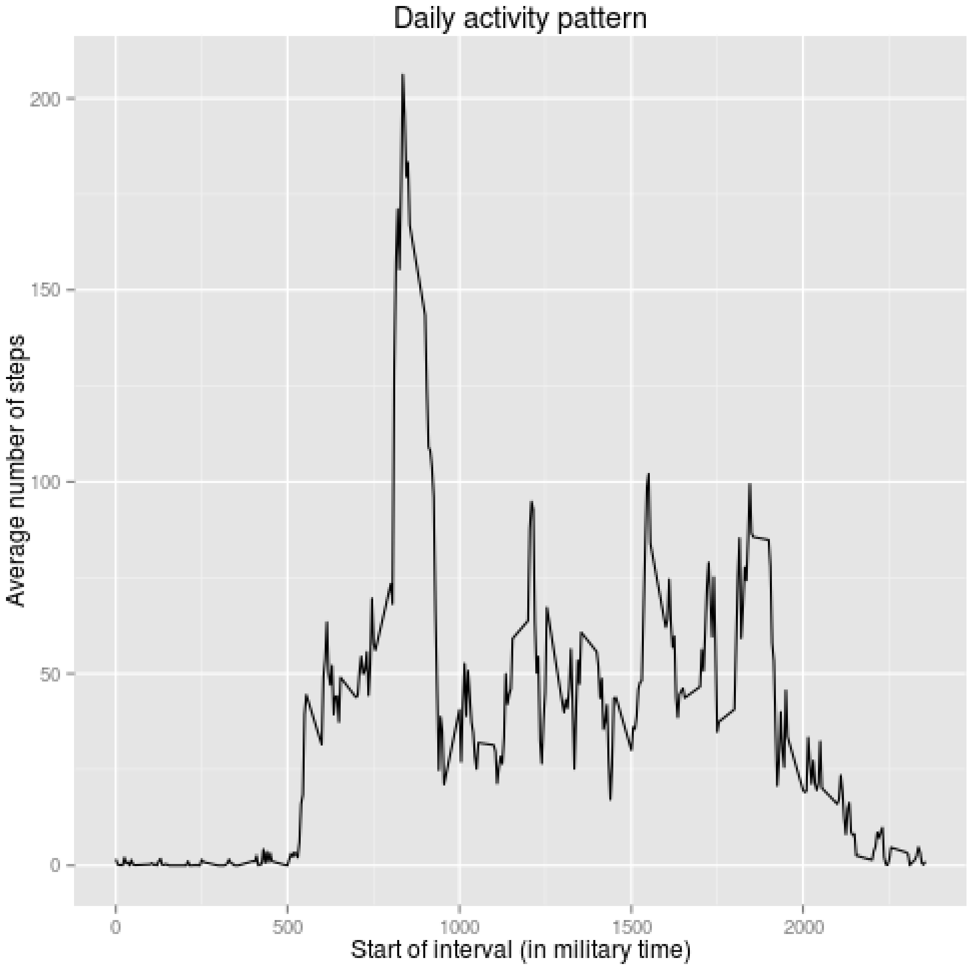
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##    41.00  8841.00 10765.00 10766.19 13294.00 21194.00
```

Daily activity pattern

Now we use `aggregate` to calculate the average number of steps per interval, and plot a time series of an average day. Step counts are near zero during typical sleeping hours and peak between 8am and 9am.

```
## Calculate average steps per interval
  intervalSteps <- aggregate(steps~interval,hasSteps,mean)

## Plot the time series
  dailyAct <- qplot(x=interval,y=steps,data=intervalSteps,geom="line")
  dailyAct <- dailyAct + labs(title="Daily activity pattern" )
  dailyAct <- dailyAct + labs(x="Start of interval (in military time)" ,
                              y="Average number of steps" )
  print(dailyAct)
```



We can see that the maximum count occurs between 8:35am and 8:40am, and the maximum is 206.17.

```
  intervalSteps$interval[which.max(intervalSteps$steps)]
```

```
## [1] 835
```

```
  max(intervalSteps$steps)
```

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

Missing values

There is missing data in this set, and that can skew the results. We can see that there are 2304 missing numbers:

```
nrow(rawData[is.na(rawData$steps),])
```

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

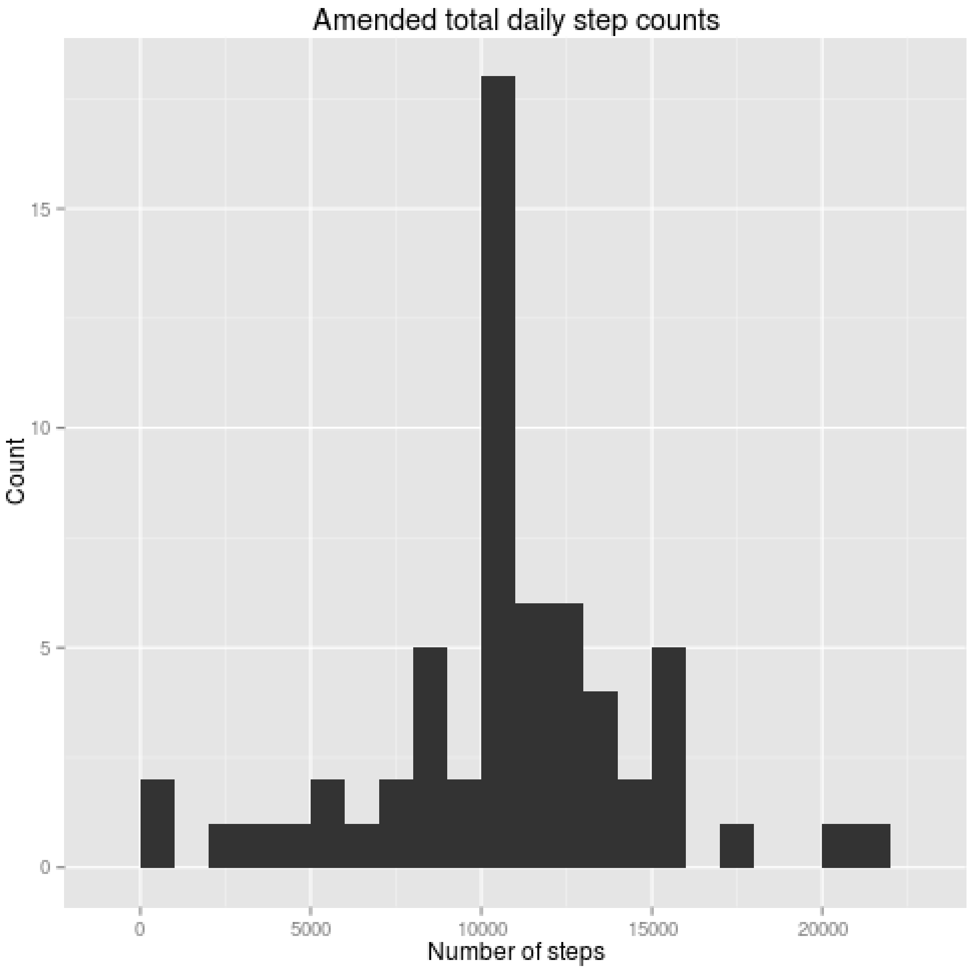
To compensate for these gaps, we fill them with the average for that given interval (assuming all intervals have at least one day for which a value was recorded).

```
  completedData <- rawData
  repIS <- rep(intervalSteps$steps,61)
  completedData$steps[is.na(completedData$steps)] <- repIS[is.na(completed
```

Now we can re-calculate the total daily step counts and re-plot the histogram:

```
  completedSteps <- aggregate(steps~date,completedData,sum)  # Calculate d

  completedHist <- qplot(completedSteps$steps, binwidth = 1000, geom="hist
  completedHist <- completedHist + labs(title="Amended total daily step co
  completedHist <- completedHist + labs(x="Number of steps", y="Count")
  print(completedHist)
```



The corrected data set has the same mean as before, but now the median is identical to the mean (at least to two decimal places).

```
  summary(completedSteps$steps,digits=7)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##    41.00  9819.00 10766.19 10766.19 12811.00 21194.00
```

Weekdays and weekends

To examine the weekday/weekend split, we create a boolean variable indicating whether the date is on the weekend, then use that to attach a new factor variable.

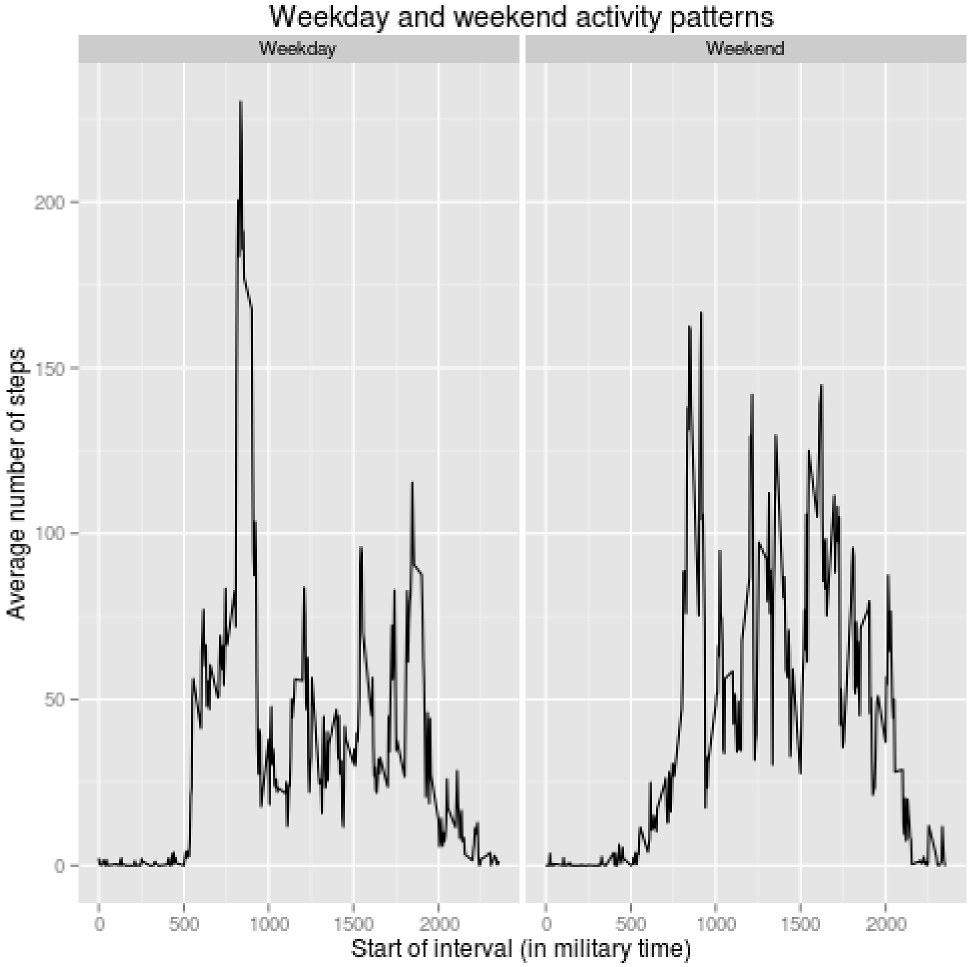
```
  onWeekend <- weekdays(completedData$date) %in% c("Saturday","Sunday")
  dayOrEnd <- rep("Weekday",length(completedData$date))
  dayOrEnd[onWeekend] <- "Weekend"
  completedData <- cbind(completedData,dayOrEnd)
```

Now we use `aggregate` to find the interval means for weekdays and for weekend days:

```
DorEintervalSteps <- aggregate(steps~interval+dayOrEnd,completedData,mea
```

Finally, we can plot the time-series for each. The sharp peak around 8:35am that we saw before is apparently much weaker on the weekends, while the step counts are higher at later hours. Clearly the typical subject has the usual morning commute and sedentary desk job.

```
deplot <- qplot(interval,steps,data=DorEintervalSteps,facets=.~dayOrEnd,
deplot <- deplot + labs(title="Weekday and weekend activity patterns")
deplot <- deplot + labs(x="Start of interval (in military time)",
                        y="Average number of steps")
print(deplot)
```


Weekday and weekend activity patterns
Weekday
Weekend
Average number of steps
Start of interval (in military time)