RR: Peer assessment 1

First the we have to aquire the data:

```
download.file("http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip","project1.zip")
unzip("project1.zip",list = T)
```

```
##           Name Length                Date
## 1 activity.csv 350829 2014-02-11 10:08:00
```

```
unzip("project1.zip")
activity.data <- read.csv("activity.csv")
```

Next, we check the structure of the data:

```
 str(activity.data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Now, we transform the date date to YMD format:

```
activity.data$date <- as.Date(activity.data$date, format= "%Y-%m- %d")
```

#What is mean total number of steps taken per day? Calculate the total number of steps taken per day:

```
 per_day_total_steps <- aggregate(steps ~ date, activity.data, sum)
 head(per_day_total_steps)
```

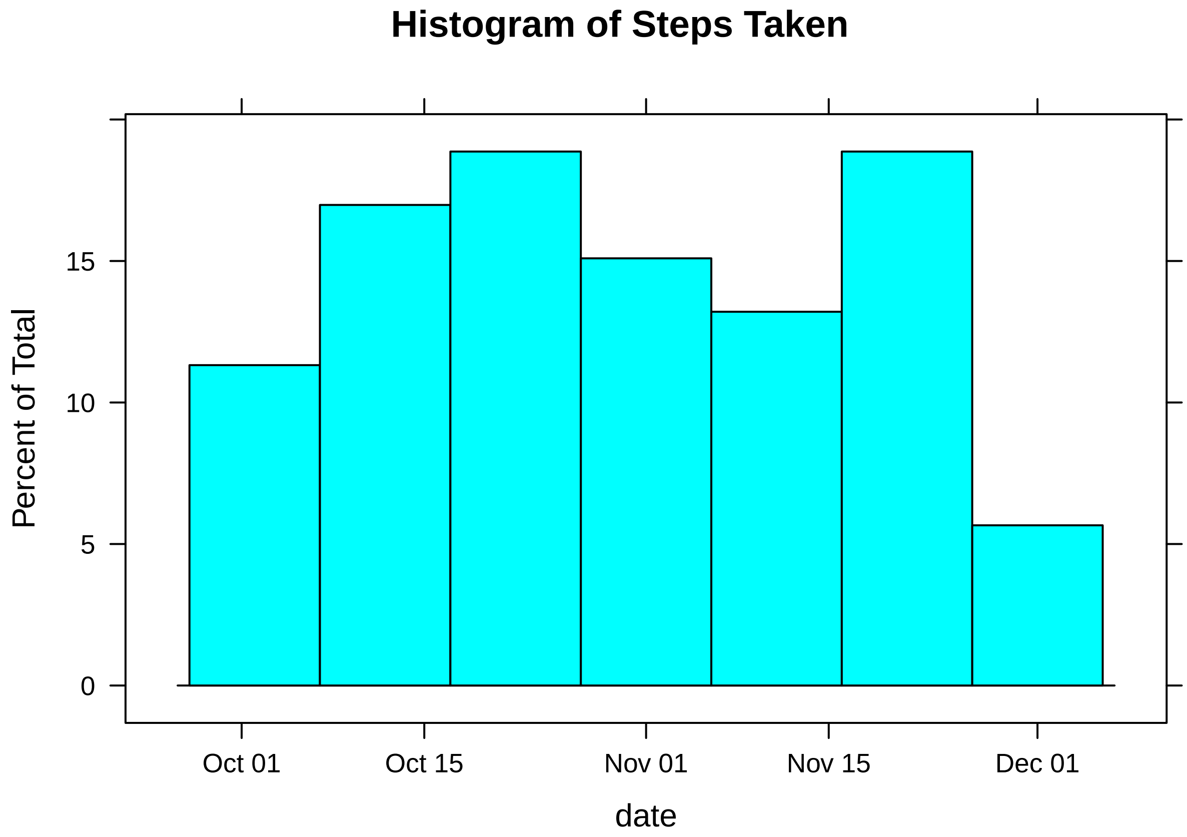
```
##         date steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

```
 tail(per_day_total_steps)
```

```
##          date steps
## 48 2012-11-24 14478
## 49 2012-11-25 11834
## 50 2012-11-26 11162
## 51 2012-11-27 13646
## 52 2012-11-28 10183
## 53 2012-11-29  7047
```

Make a histogram of the total number of steps taken each day:

```
lattice::histogram(steps~date, per_day_total_steps, breaks = 6, main = "Histogram of Steps Taken")
```



```
mean(per_day_total_steps$steps, na.rm = T)
```

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

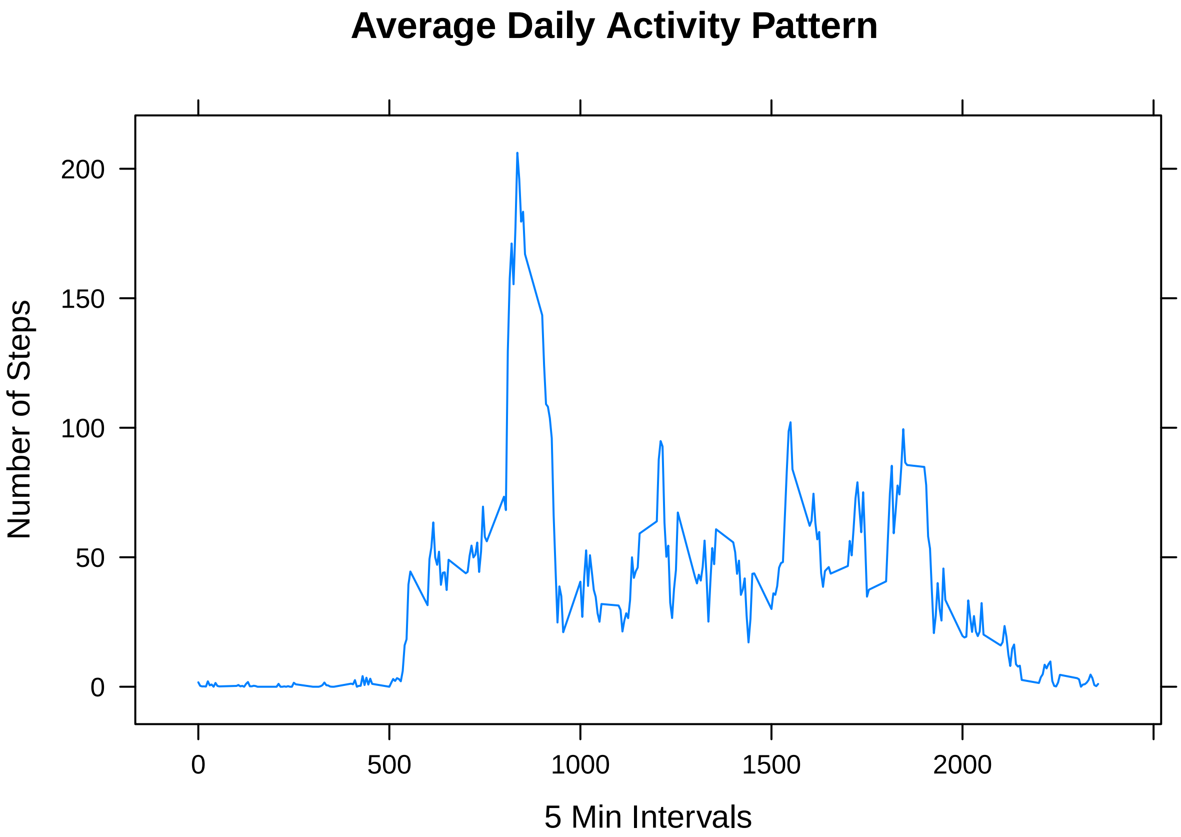
```
median(per_day_total_steps$steps, na.rm = T)
```

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

What is the average daily activity pattern?

Aggregrate the steps by intervals and plot

```
interval_steps <- aggregate(activity.data$steps, by= list(interval= activity.data$interval), FUN= mean,
lattice::xyplot(x ~ interval, data=interval_steps, type= "l", xlab="5 Min Intervals", ylab="Number of S
```



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

```
psych::describe(interval_steps)
```

```
##          vars   n    mean     sd  median trimmed    mad min     max
## interval    1 288 1177.50 693.64 1177.50 1177.50 889.56   0 2355.00
## x           2 288   37.38  38.66   34.11   31.34  38.86   0  206.17
##            range skew kurtosis    se
## interval 2355.00 0.00    -1.22 40.87
## x         206.17 1.66     3.74  2.28
```

```
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
```

```
interval_steps %>% filter(x>200.00)
```

```
##   interval        x
## 1      835 206.1698
```

Imputing missing values

Calculate and report the total number of missing values in the dataset:

```
sum(is.na(activity.data$steps))
```

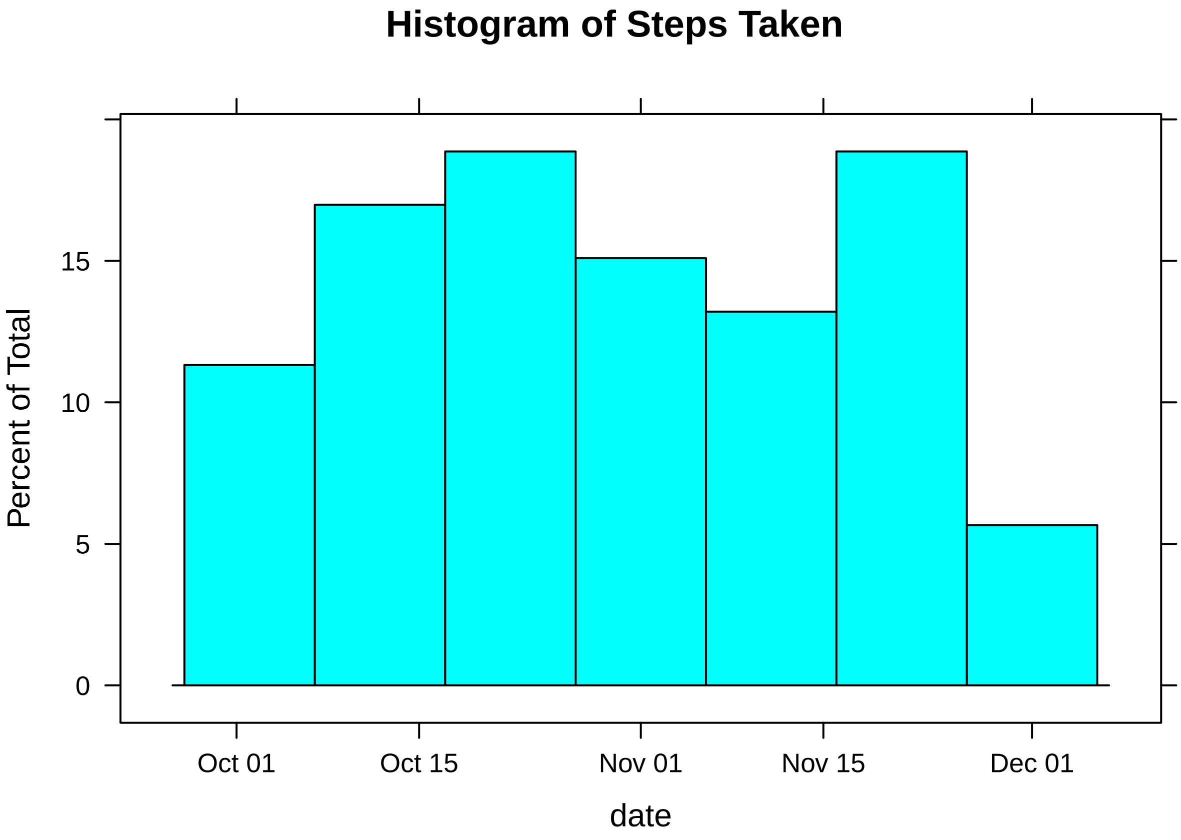
```
## [1] 2304
```

Devise a strategy for filling in all of the missing values in the dataset and Create a new dataset that is equal to the original dataset but with the missing data filled in:

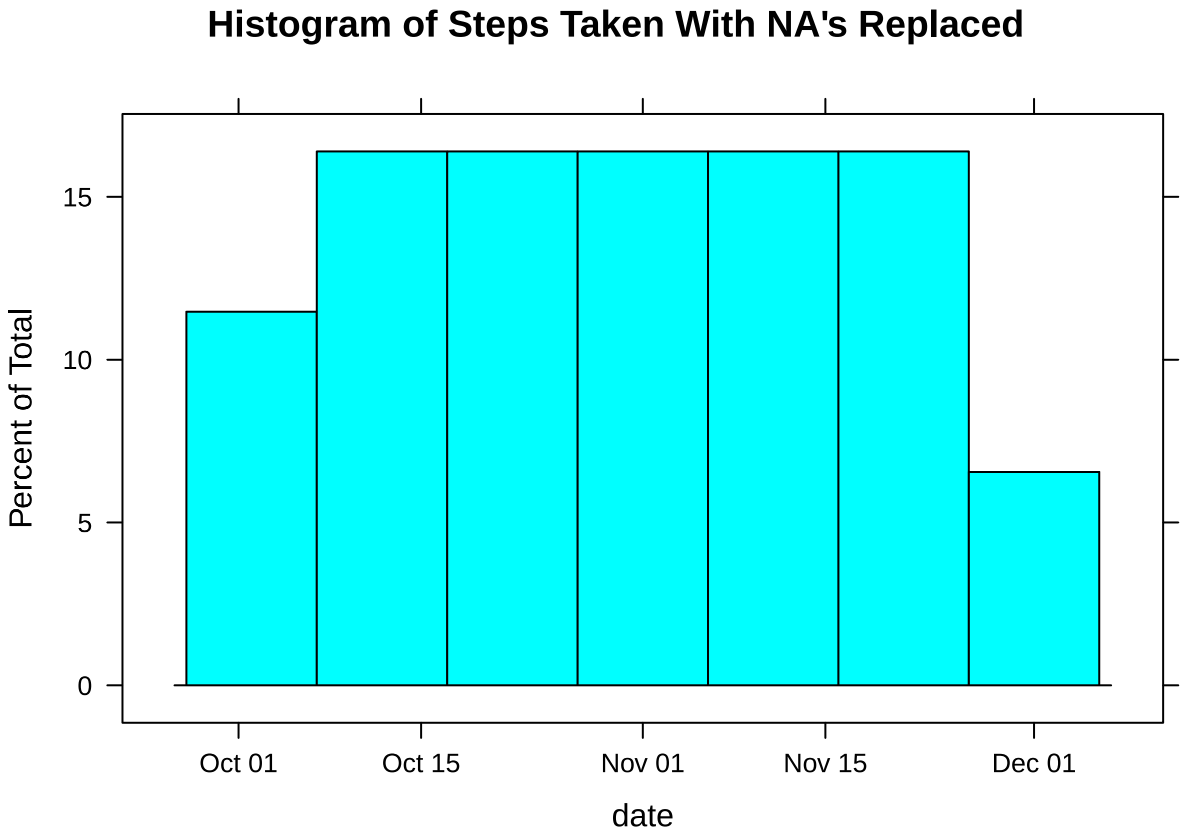
```
activity.data_filled <- gam::na.gam.replace(activity.data)
```

Make a 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:

```
lattice::histogram(steps~date, per_day_total_steps, breaks = 6, main = "Histogram of Steps Taken")
```



```
lattice::histogram(steps~date, activity.data_filled, breaks = 6, main = "Histogram of Steps Taken With
```



We can see that the new data set is different than the one with NA's, and maybe the gam package is not the best for this:

```
mean(activity.data_filled$steps)
```

```
## [1] 37.3826
```

```
median(activity.data_filled$steps)
```

```
## [1] 0
```

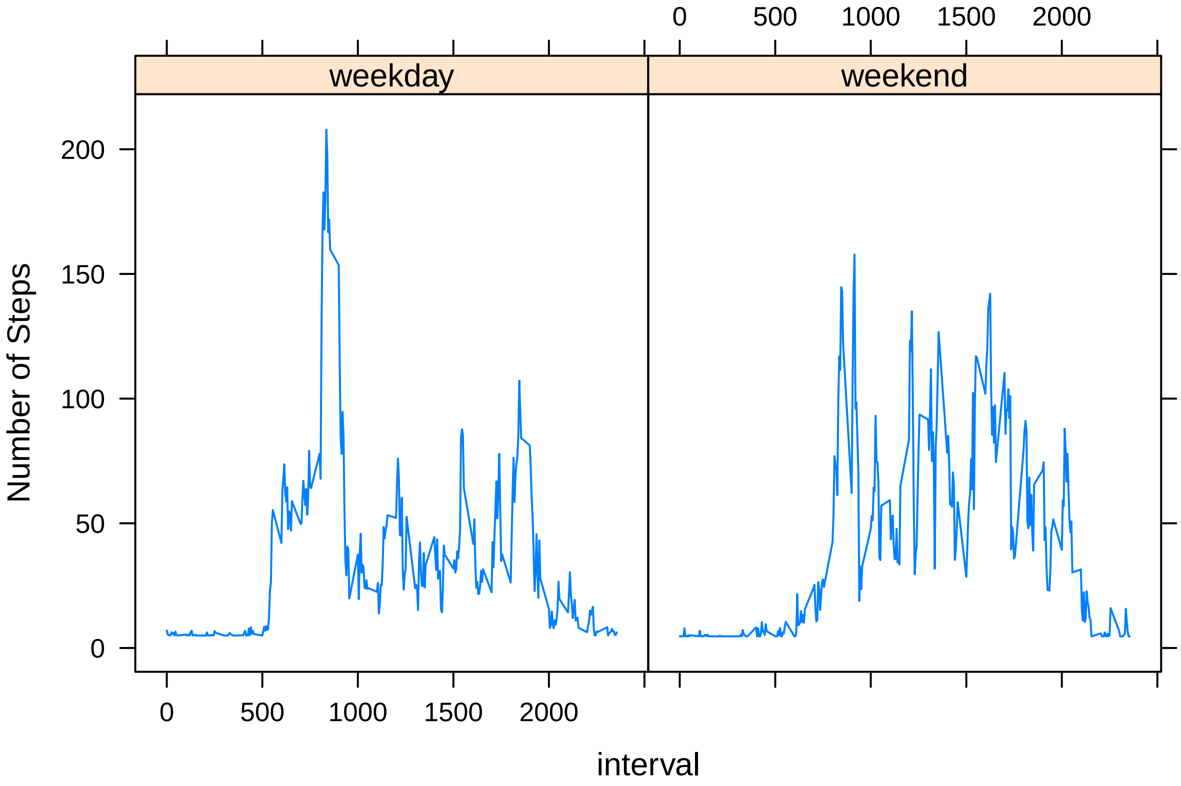
Are there differences in activity patterns between weekdays and weekends?

```
weekday_or_end <- function(date) {
    day <- weekdays(date)
    if (day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))
        return("weekday") else if (day %in% c("Saturday", "Sunday"))
        return("weekend") else stop("invalid date")
}
```

```
activity.data_filled$date <- as.Date(activity.data_filled$date)
activity.data_filled$day <- sapply(activity.data_filled$date,weekday_or_end)
```

Now, let's make a panel plot containing plots of average number of steps taken on weekdays and weekends.

```
avgs <- aggregate(steps ~ interval + day, data = activity.data_filled, mean)
lattice::xyplot(steps ~ interval| as.factor(day), data=avgs, type="l", ylab ="Number of Steps")
```


weekday
weekend
Number of Steps
interval