Reproducible Research: Project 1

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

```
library(ggplot2)
library(scales)
library(gridExtra)
```

```
## Loading required package: grid
```

```
data <- read.csv(file="activity.csv")
str(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 ...
```

```
data$date <- as.Date(data$date)
summary(data)
```

```
##      steps             date               interval
##  Min.   :  0.0   Min.   :2012-10-01   Min.   :   0
##  1st Qu.:  0.0   1st Qu.:2012-10-16   1st Qu.: 589
##  Median :  0.0   Median :2012-10-31   Median :1178
##  Mean   : 37.4   Mean   :2012-10-31   Mean   :1178
##  3rd Qu.: 12.0   3rd Qu.:2012-11-15   3rd Qu.:1766
##  Max.   :806.0   Max.   :2012-11-30   Max.   :2355
##  NA's   :2304
```

```
steps.per.day <- as.data.frame(rowsum(data$steps, data$date))
steps.per.day$date <- as.Date(rownames(steps.per.day))
rownames(steps.per.day) <- NULL
colnames(steps.per.day) <- c("steps", "date")
```

```
##What is the mean total of steps taken per day?
```

```
mean(steps.per.day$steps, na.rm=TRUE)
```

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

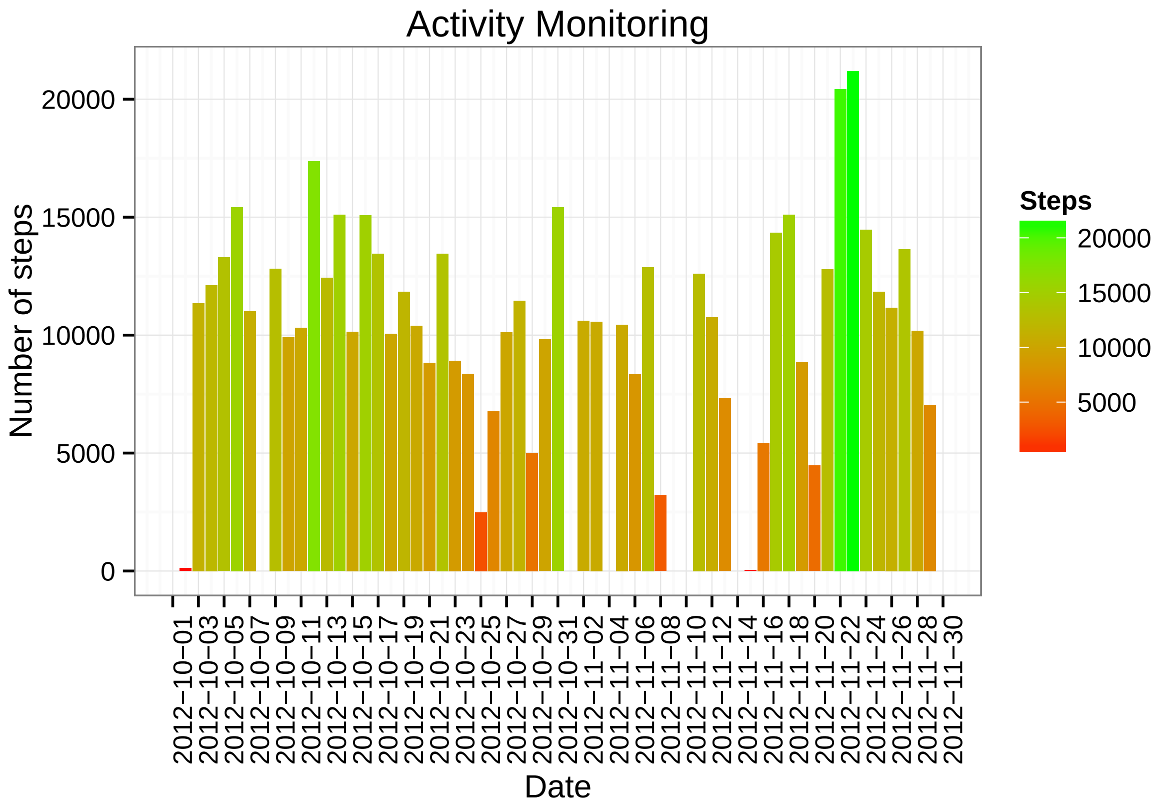
```
##What is the median total number of steps taken per day
```

```
median(steps.per.day$steps, na.rm=TRUE)
```

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

```
##Make a histogram of the total number of stpes taken each day.
```

```
Histogram1 <- ggplot(steps.per.day, aes(x=date, y=steps)) +
            geom_histogram(stat="identity",
                           binwidth=nrow(steps.per.day),
                           position="identity",
                           aes(fill=steps,)) +
            scale_fill_gradient("Steps", low = "red", high = "green") +
            scale_x_date(labels = date_format("%Y-%m-%d"),
                         breaks = seq(min(steps.per.day$date),
                                      max(steps.per.day$date),
                                      length=ceiling(nrow(steps.per.day)/2)),
                         limits = c(min(steps.per.day$date),
                                    max(steps.per.day$date))) +
            labs(title = "Activity Monitoring") +
            labs(x = "Date", y = "Number of steps") +
            theme_bw() +
            theme(axis.text.x = element_text(angle = 90, hjust = 1))
Histogram1
```



```
##Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of
```

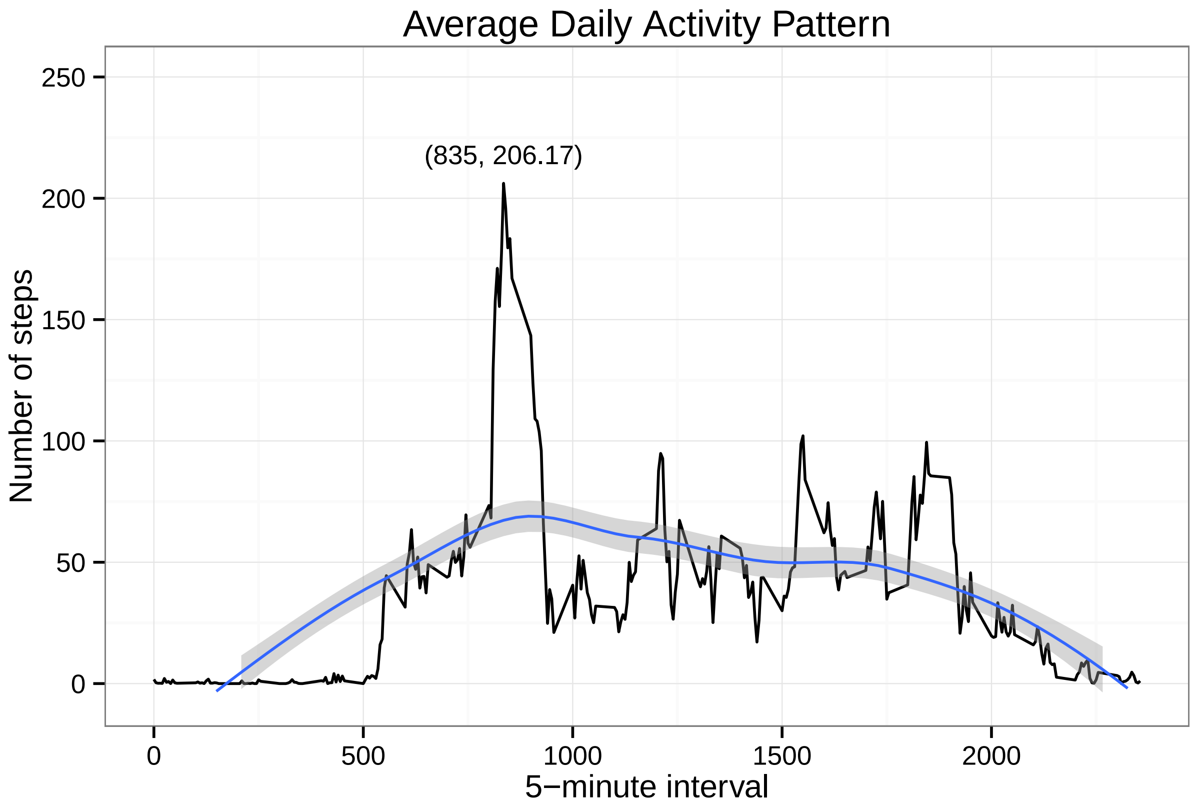
```
time.series.plot <- data.frame(steps=data$steps,
                               interval=data$interval)
time.series.plot$interval <- as.factor(time.series.plot$interval)

time.series.plot<- aggregate(steps ~ interval, time.series.plot, mean)
```

```r
time.series.plot$interval <- as.numeric(levels(time.series.plot$interval))[time.series.plot$interval]
maxID <- which.max(time.series.plot$steps)

Histogram2 <- ggplot(time.series.plot, aes(x=interval, y=steps)) +
            geom_line() +
            geom_smooth(method="loess") +
            geom_text(data=time.series.plot[maxID, ],
                      label=sprintf("(%i, %.2f)",
                                    time.series.plot[maxID,]$interval,
                                    time.series.plot[maxID,]$steps),
                      size=3.4,
                      vjust=-1,) +
            scale_y_continuous(limits = c(-5, 250)) +
            labs(title = "Average Daily Activity Pattern") +
            labs(x = "5-minute interval",
                 y = "Number of steps") +
            theme_bw()
Histogram2
```

```
## Warning: Removed 6 rows containing missing values (geom_path).
```



```
##Which of the 5-minute interval, on average across all the days in the dataset, contains the maximum nu
```

```r
time.series.plot[maxID,]
```

```
##     interval steps
## 104      835 206.2
```

```
##What is the total number of missing values in the dataset (i.e. the total number of rows with NAs)?
```

```
selectNA <- !complete.cases(data$steps)
selectZero <- data$steps == 0
print(sprintf("Total number of missing values in the dataset: %i", sum(selectNA)))
```

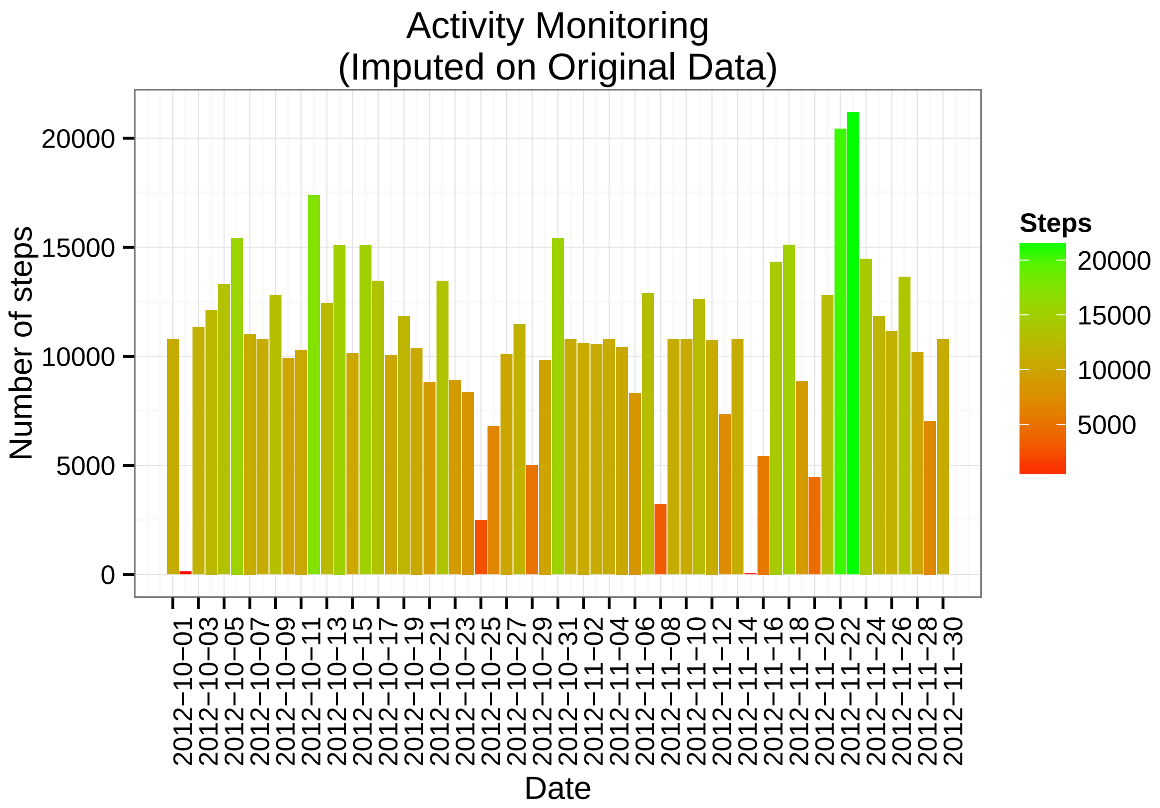
```
## [1] "Total number of missing values in the dataset: 2304"
```

```
##What could be a strategy for filling in all of the missing values in the dataset? We you could use mea
```

```
dataImputed <- data
estimator <- mean(dataImputed$steps, na.rm=TRUE)
dataImputed$steps[selectNA] <- estimator
```

```
steps.per.dayImputed <- as.data.frame(rowsum(dataImputed$steps, dataImputed$date))
steps.per.dayImputed$date <- as.Date(rownames(steps.per.dayImputed))
rownames(steps.per.dayImputed) <- NULL
colnames(steps.per.dayImputed) <- c("steps", "date")

Histogram3 <- ggplot(steps.per.dayImputed, aes(x=date, y=steps)) +
            geom_histogram(stat="identity",
                           binwidth=nrow(steps.per.dayImputed),
                           position="identity",
                           aes(fill=steps,)) +
            scale_fill_gradient("Steps", low = "red", high = "green") +
            scale_x_date(labels = date_format("%Y-%m-%d"),
                         breaks = seq(min(steps.per.dayImputed$date),
                                      max(steps.per.dayImputed$date),
                                      length=ceiling(nrow(steps.per.dayImputed)/2)),
                         limits = c(min(steps.per.dayImputed$date),
                                    max(steps.per.dayImputed$date))) +
            labs(title = "Activity Monitoring\n(Imputed on Original Data)") +
            labs(x = "Date", y = "Number of steps") +
            theme_bw() +
            theme(axis.text.x = element_text(angle = 90, hjust = 1))
Histogram3
```



```
##What then are new datasets equal to the original dataset but with the missing data filled in?
```

```
unique(dataImputed$date[selectNA])
```

```
## [1] "2012-10-01" "2012-10-08" "2012-11-01" "2012-11-04" "2012-11-09"
## [6] "2012-11-10" "2012-11-14" "2012-11-30"
```

```
summary(steps.per.day)
```

```
##      steps             date
##  Min.   :   41   Min.   :2012-10-01
##  1st Qu.: 8841   1st Qu.:2012-10-16
##  Median :10765   Median :2012-10-31
##  Mean   :10766   Mean   :2012-10-31
##  3rd Qu.:13294   3rd Qu.:2012-11-15
##  Max.   :21194   Max.   :2012-11-30
##  NA's   :8
```

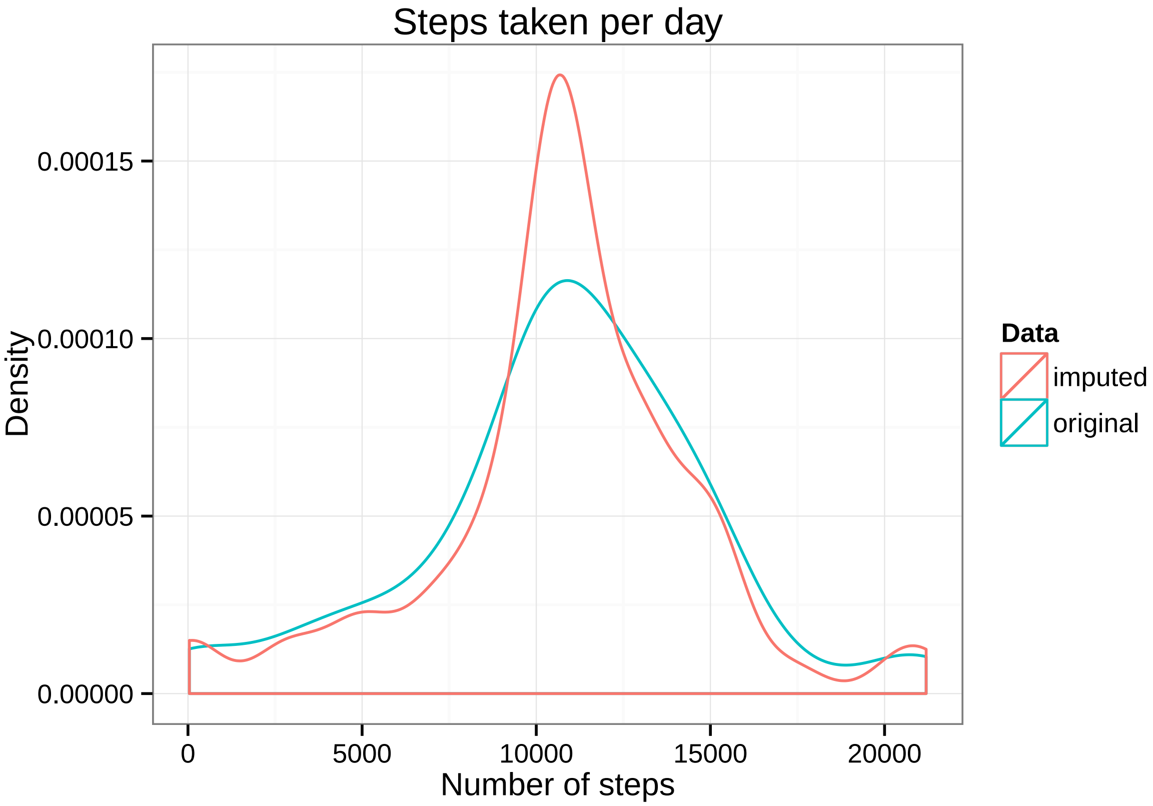
```
summary(steps.per.dayImputed)
```

```
##      steps             date
##  Min.   :   41   Min.   :2012-10-01
##  1st Qu.: 9819   1st Qu.:2012-10-16
##  Median :10766   Median :2012-10-31
##  Mean   :10766   Mean   :2012-10-31
##  3rd Qu.:12811   3rd Qu.:2012-11-15
##  Max.   :21194   Max.   :2012-11-30
```

```
##Make a histogram of the total number of steps taken each day and Calculate and report the mean and med
```

```
Histogram4 <- ggplot() +
            geom_density(data=steps.per.day, aes(x=steps,
                                                   y=..density..,
                                                   color="original"),
                         na.rm=TRUE) +
            geom_density(data=steps.per.dayImputed, aes(x=steps,
                                                          y=..density..,
                                                          color="imputed")) +
            scale_color_discrete(name ="Data", labels=c("imputed", "original")) +
            labs(x="Number of steps", y="Density") +
            labs(title="Steps taken per day") +
            theme_bw()
Histogram4
```

```
## Warning: Removed 8 rows containing non-finite values (stat_density).
```



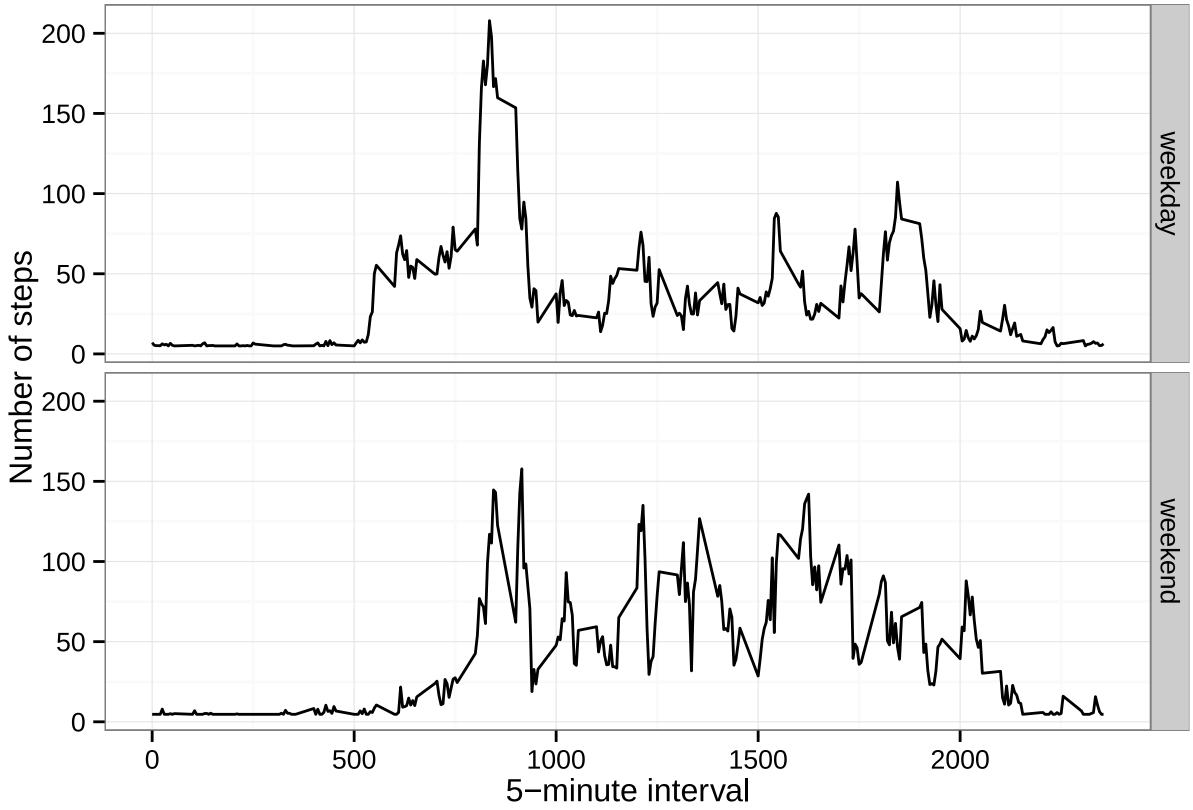
```
##Notice from the histogram that the imputed values are noticeably higher than the original values. How
```

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

```
time.series.dataImputed <- dataImputed
time.series.dataImputed$interval <- as.factor(time.series.dataImputed$interval)
day <- !weekdays(time.series.dataImputed$date) %in% c("Saturday", "Sunday")
day[day == TRUE] <- "weekday"
```

```
day[day == FALSE] <- "weekend"
day <- as.factor(day)

time.series.dataImputed$day <- day
time.series.dataImputed <- aggregate(steps ~ interval + day, time.series.dataImputed, mean)
time.series.dataImputed$interval <- as.numeric(levels(time.series.dataImputed$interval))[time.series.da
Histogram5 <- ggplot(time.series.dataImputed, aes(interval, steps)) +
        geom_line() +
        facet_grid(day ~ .) +
        labs(x="5-minute interval",
             y="Number of steps") +
        theme_bw()
Histogram5
```



```
##Weekends seem to be of higher intensity compared to weekdays.
```