Loading the data.

```
temp <- tempfile()
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip",temp)
Activity.monitoring.data <- read.csv(unz(temp, "activity.csv"), header = TRUE,colClasses = c("nu
meric", "Date", "numeric"))
unlink(temp)
```

Calculate the total number of steps taken per day.

```
clean_activity <- na.omit(Activity.monitoring.data)
head(clean_activity)
```

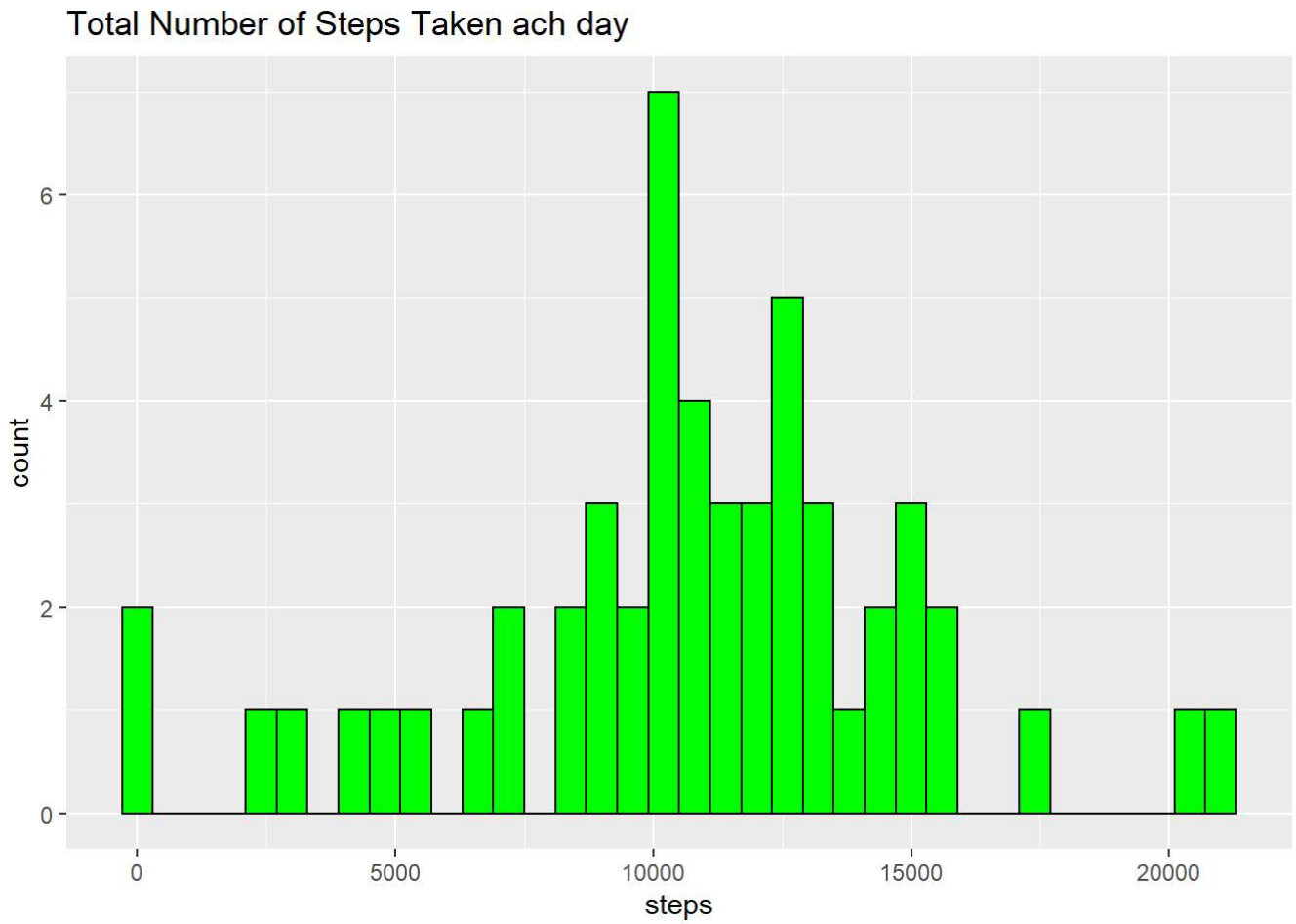
```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
```

```
total.steps <- aggregate(steps~date, clean_activity, sum)
head(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
```

Histogram of the total number of steps taken each day.

```
library(ggplot2)
ggplot(data = total.steps) + geom_histogram(aes(steps), binwidth = 600, fill = 'green', col = 'b
lack') + ggtitle("Total Number of Steps Taken ach day")
```



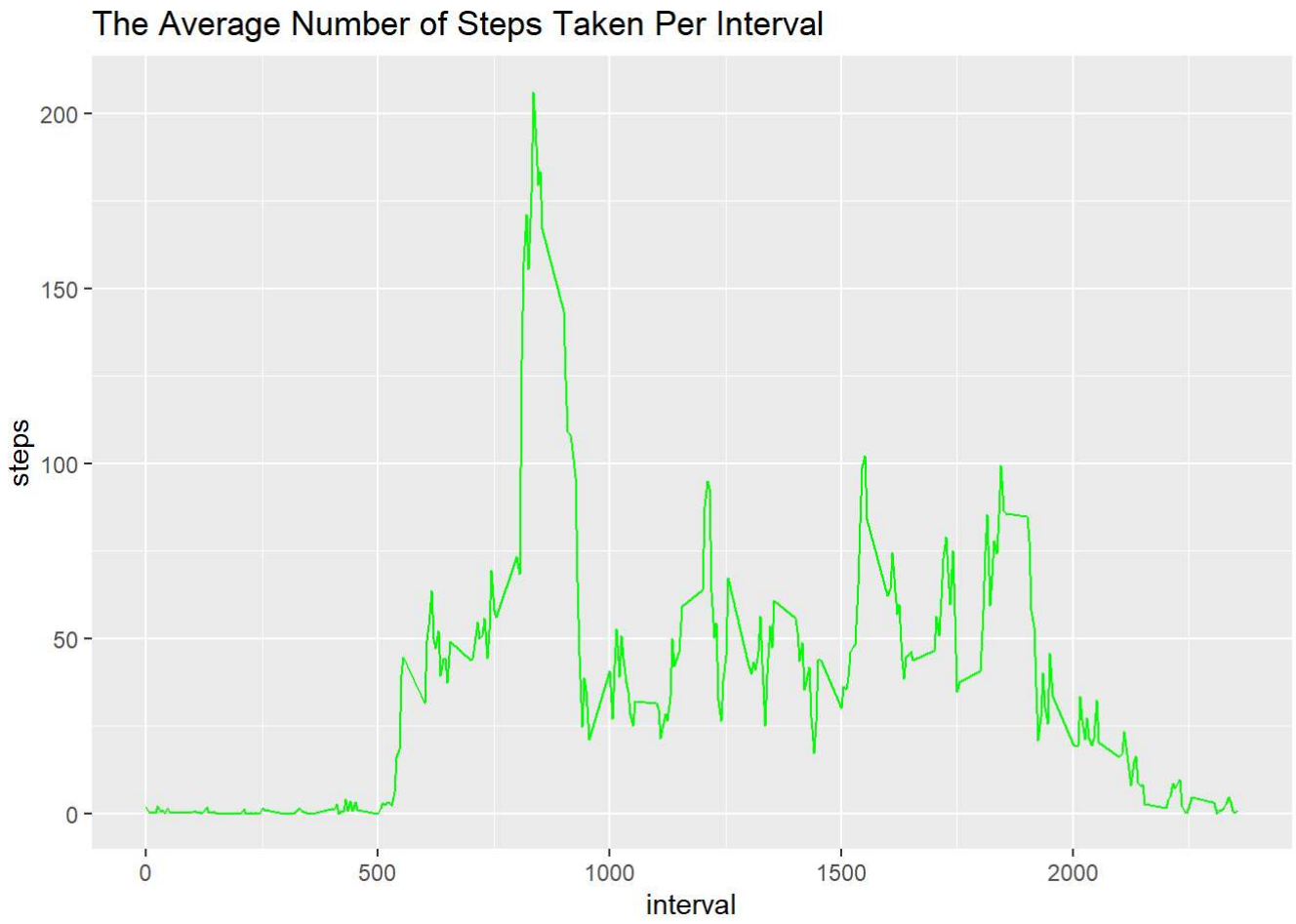
Calculate and report the mean and median of the total number of steps taken per day.

```
summary(total.steps)
```

```
##       date               steps
##  Min.   :2012-10-02   Min.   :   41
##  1st Qu.:2012-10-16   1st Qu.: 8841
##  Median :2012-10-29   Median :10765
##  Mean   :2012-10-30   Mean   :10766
##  3rd Qu.:2012-11-16   3rd Qu.:13294
##  Max.   :2012-11-29   Max.   :21194
```

Time series plot (i.e.type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis).

```
total.stepszz <- aggregate(steps~interval, Activity.monitoring.data, mean)
ggplot(data = total.stepszz) + geom_line(color = "green", size = 0.5, (aes(interval,steps))) + g
gtitle("The Average Number of Steps Taken Per Interval")
```



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

```
total.stepszz[which.max(total.stepszz$steps),]
```

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

The total number of missing values in the dataset.

```
colSums(is.na(Activity.monitoring.data))
```

```
##    steps     date interval
##     2304        0        0
```

Strategy for filling in all of the missing values in the dataset.

```
Filled_data <- total.stepszz$mean[match(Activity.monitoring.data$interval, total.stepszz$interva
l)]
```

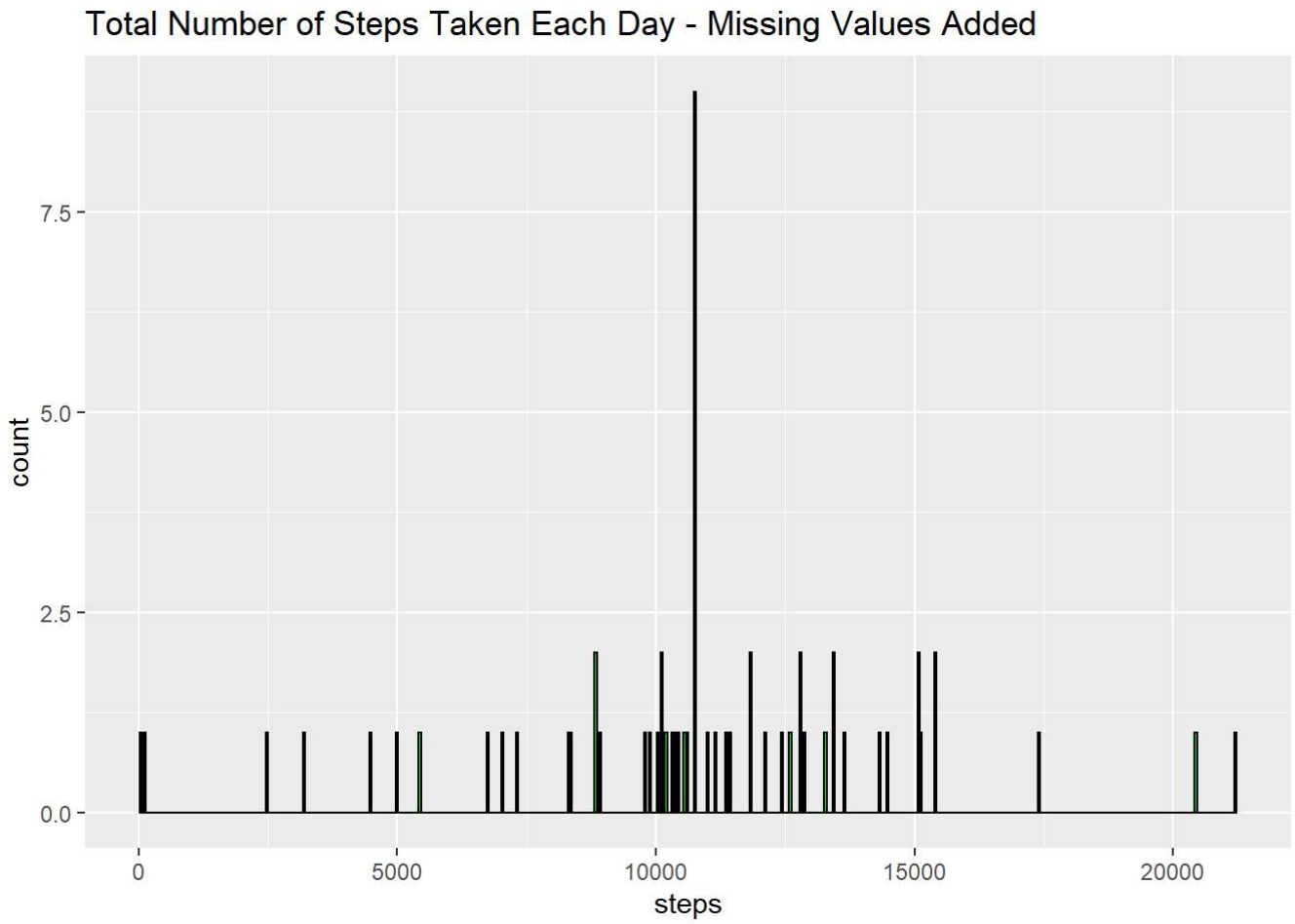
A new dataset that is equal to the original dataset but with the missing data filled in.

```
Activity.Filled <- Activity.monitoring.data
Activity.Filled$steps <- ifelse(is.na(Activity.Filled$steps) == TRUE, total.stepszz$steps[total.
stepszz$interval %in% Activity.Filled$interval], Activity.Filled$steps)
head(Activity.Filled)
```

```
##       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
```

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.

```
Activity.Filledv2 <- aggregate(steps~date, Activity.Filled, sum)
ggplot( data = Activity.Filledv2) + geom_histogram(aes(steps), fill = "green", col = "black", bi
nwidth = 40) + ggtitle("Total Number of Steps Taken Each Day - Missing Values Added")
```




```
summary(Activity.Filledv2)
```

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

Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

Yes, this is due to replacing the NA values with the average values. The impact of the missing data in small. The median for example is only about 1 step higher.

new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
activity.new <- Activity.monitoring.data
library(data.table)
activity.new <- data.table(activity.new)
activity.new[, date := as.POSIXct(date, format = "%Y-%m-%d")]
activity.new[, `Day of Week`:= weekdays(x = date)]
activity.new[grepl(pattern = "Monday|Tuesday|Wednesday|Thursday|Friday", x = `Day of Week`), "we
ekday or weekend"] <- "weekday"
activity.new[grepl(pattern = "Saturday|Sunday", x = `Day of Week`), "weekday or weekend"] <- "we
ekend"
activity.new[, `weekday or weekend` := as.factor(`weekday or weekend`)]
head(activity.new, 10)
```

```
##     steps                date interval Day of Week weekday or weekend
##  1:    NA 2012-09-30 20:00:00        0      Sunday            weekend
##  2:    NA 2012-09-30 20:00:00        5      Sunday            weekend
##  3:    NA 2012-09-30 20:00:00       10      Sunday            weekend
##  4:    NA 2012-09-30 20:00:00       15      Sunday            weekend
##  5:    NA 2012-09-30 20:00:00       20      Sunday            weekend
##  6:    NA 2012-09-30 20:00:00       25      Sunday            weekend
##  7:    NA 2012-09-30 20:00:00       30      Sunday            weekend
##  8:    NA 2012-09-30 20:00:00       35      Sunday            weekend
##  9:    NA 2012-09-30 20:00:00       40      Sunday            weekend
## 10:    NA 2012-09-30 20:00:00       45      Sunday            weekend
```

Panel plot containing a time series plot (i.e.type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

```
activity.new[is.na(steps), "steps"] <- activity.new[, c(lapply(.SD, median, na.rm = TRUE)), .SDc
ols = c("steps")]

activity.new.Interval <- activity.new[, c(lapply(.SD, mean, na.rm = TRUE)), .SDcols = c("steps"
), by = .(interval, `weekday or weekend`)]

library(ggplot2)

ggplot(activity.new.Interval, aes(x = interval , y = steps, color=`weekday or weekend`,  binwidt
h = 600)) + geom_line(color = "green") + labs(title = "Avg. Daily Steps by Weektype", x = "Inter
val", y = "No. of Steps") + facet_wrap(~`weekday or weekend` , ncol = 1, nrow=2)
```